

A wide-angle photograph of a lush green hillside under a blue sky with scattered clouds. In the foreground, a single sheep grazes on the grass. The hillside slopes down towards a valley in the distance.

Improving the transparency and usability of health economic models

HERC, University of Oxford | Dr. Robert Smith | July 2024

 rsmith@darkpeakanalytics.com

 <https://github.com/dark-peak-analytics>

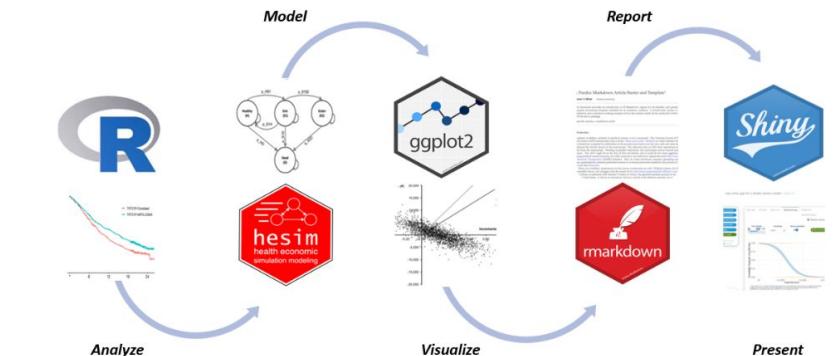
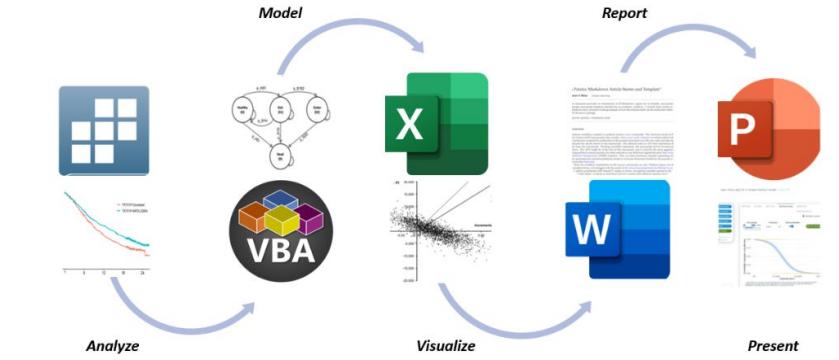
 <https://www.linkedin.com/company/dark-peak-analytics>

A group of nine dogs of various breeds and colors are gathered outside a wooden building. In the foreground, a white greyhound sits on the left, looking towards the right. Next to it is a black and tan cocker spaniel. Behind them are several other dogs, including a small white dog, a black and white dog, a large black dog, a brown and white dog, a light-colored curly-haired dog, and a larger tan dog. A white sign with the words "POLLING STATION" in bold black capital letters is mounted on the wooden railing of the entrance. The scene is set against a backdrop of green bushes and trees.

POLLING
STATION

Overarching aim

Shifting the modelling pipeline from spreadsheet software (e.g. MS Excel) to script-based programming languages (e.g R).



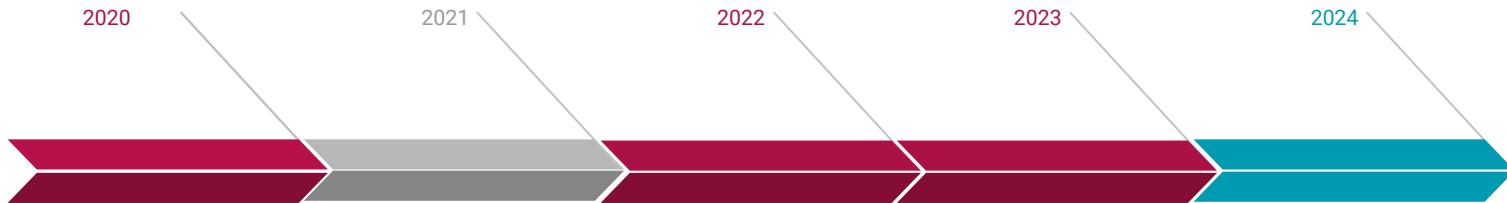
MS Excel vs R (python / others)



			Comments	Link to Slide
1	Familiarity Barriers to entry	● ● ● ●	●	Coding in R is hard Teaching
2	Capabilities	● ●	● ● ● ● ●	... but much more powerful Comparison
3	Computational Speed	●	● ● ● ●	... and faster Benchmark
4	Development & Iteration speed	● ●	● ● ● ●	... and easier to develop at scale Living HTA
5	Transparency	● ●	● ● ● ● ●	... and more robust (int. & ext. review) Packages; Data/model
6	Engagement & Visualisation	● ● ●	● ● ● ●	... and looks better Shiny UI

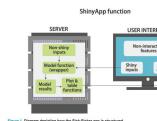
An entirely personal, and very biased set of estimates!

Related publications



Making Health Economic Models Shiny: A tutorial

Smith RA and Schneider PP. Making health economic models Shiny: A tutorial. *Wellcome Open Res* 2020, **5**:69 (<https://doi.org/10.12688/wellcomeopenres.15807.2>)



covid-19 work

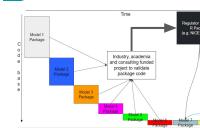
Living HTA: Automating Health Economic Evaluation with R

Smith RA, Schneider PP and Mohammed W. Living HTA: Automating Health Economic Evaluation with R. *Wellcome Open Res* 2022, **7**:194 (<https://doi.org/10.12688/wellcomeopenres.17933.2>)



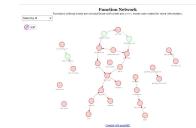
Packaging cost-effectiveness models in R: A tutorial.

Smith RA, Mohammed W and Schneider PP. R Packaging cost-effectiveness models in R: A tutorial. 2023. (<https://wellcomeopenresearch.org/articles/8-419>)



assertHE: an R package to improve quality assurance of health economic models

Smith et al. assertHE: an R package to improve quality assurance of health economic models. 2024. (<https://github.com/dark-peak-analytics/assertHE>)

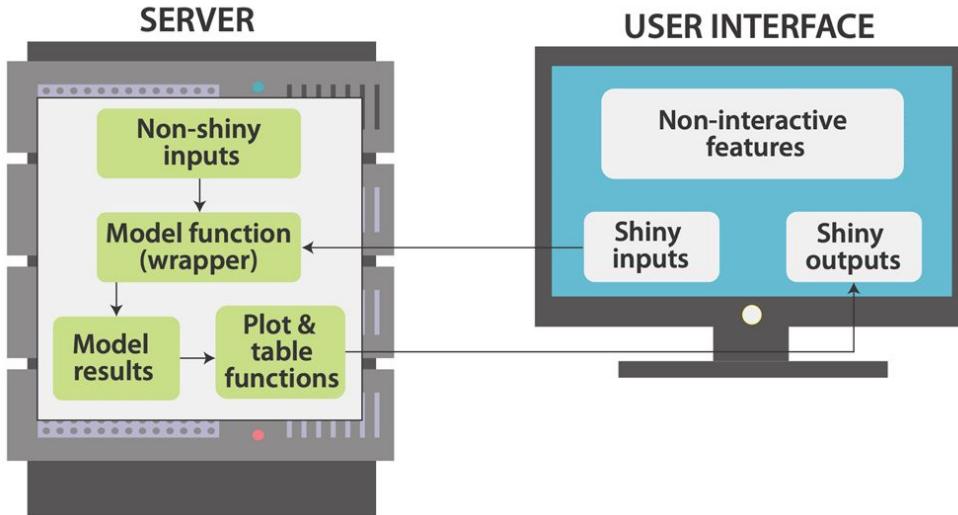


Making Health Economic Models Shiny



Making Health Economic Models Shiny

ShinyApp function



Wellcome Open Research | Wellcome Open Research 2020, 5:63 Last updated: 05/03/2022 | Check for updates

METHOD ARTICLE | Review | Making health economic models Shiny: A tutorial [version 2; peer review: 2 approved]

Robert A. Smith , Paul Schneider 

School of Health and Related Research, University of Sheffield, Regents Court, Sheffield, S1 4DA, UK
* Equal contributors.

V2 First published: 14 Apr 2020, 5:69 <https://doi.org/10.12688/wellcomeresources.15807.1> Latest published: 31 Jul 2020, 5:69 <https://doi.org/10.12688/wellcomeresources.15807.2>

Open Peer Review

Approval Status	1	2
version 2 (revision)		
version 1 (revision)		

1. Talitha L. Feenstra, University of Groningen, Groningen, The Netherlands
2. Yiqiao Xin , University of Glasgow, Glasgow, UK

Any reports and responses or comments on the article can be found at the end of the article.

Abstract
Health economic evaluation models have traditionally been built in Microsoft Excel, but more sophisticated tools are increasingly being used to build these models. Of all the programming languages, R is most popular amongst health economists because it has a plethora of user created packages and is highly flexible. However, even with an integrated development environment such as R Studio, R lacks a simple point-and-click user interface, which requires some programming knowledge. This might make the switch from Microsoft Excel to R seem daunting, and it might make it difficult to directly communicate results with decision makers and other stakeholders.

The R package Shiny has the potential to resolve this limitation. It allows programmers to embed health economic models developed in R into interactive web browser based user interfaces. Users can specify their own parameters and run different scenarios. In a Markov model, can be computed within seconds. This paper provides a tutorial on how to wrap a health economic model built in R into a Shiny application. We use a four-state Markov model developed by the Decision Analysis in R for Technologies in Health (DARTH) group as a case study to demonstrate main principles and basic functionality.

A more extensive tutorial, all code, and data are provided in a [GitHub repository](#).

Keywords
Health Economics, R, RShiny, Decision Science

Page 1 of 26



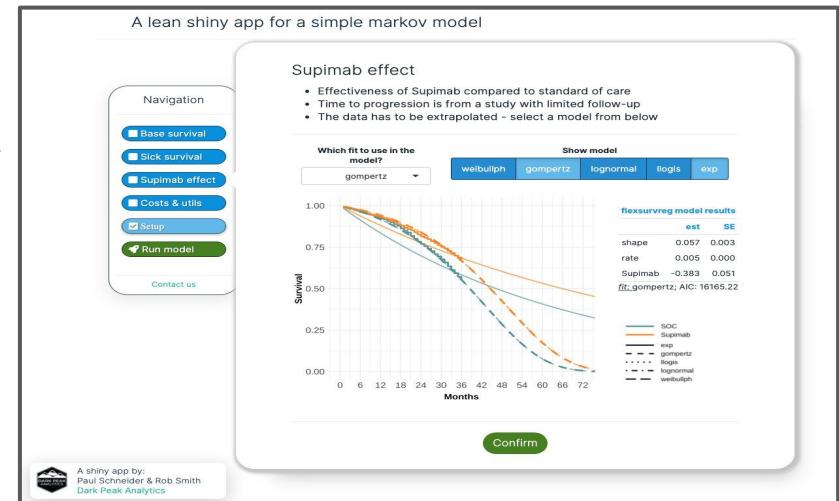
Making Health Economic Models Shiny

GitHub/CRAN



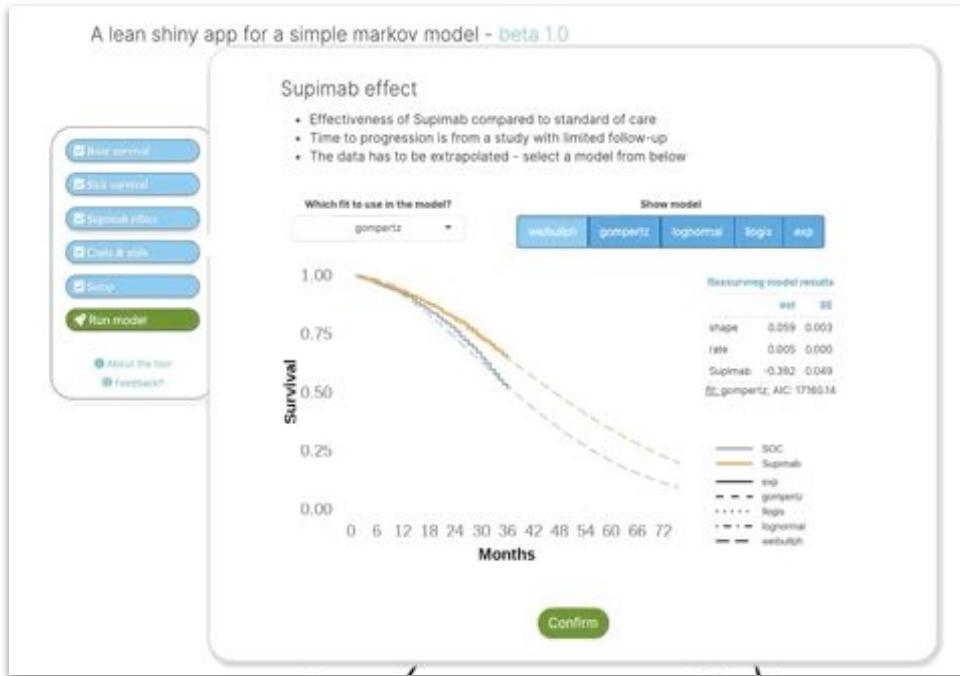
A faster method of
running time
dependent markov
models using C++

<https://github.com/dark-peak-analytics/darkpeak>



<https://darkpeakanalytics.shinyapps.io/sadm-mk2/>

Making Health Economic Models Shiny

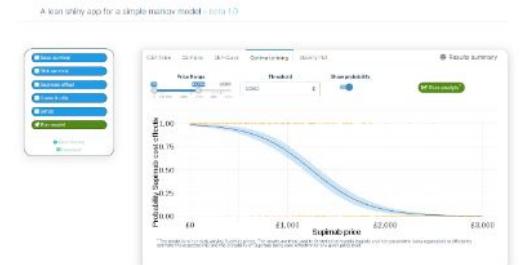


<https://darkpeakanalytics.shinyapps.io/sadm-mk2/>

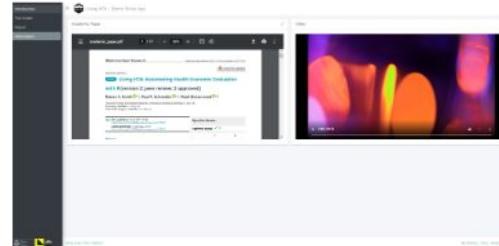
Making Health Economic Models Shiny



<https://shiny.york.ac.uk/dceasimple/>



<https://darkpeakanalytics.shinyapps.io/sadm-mk2/>

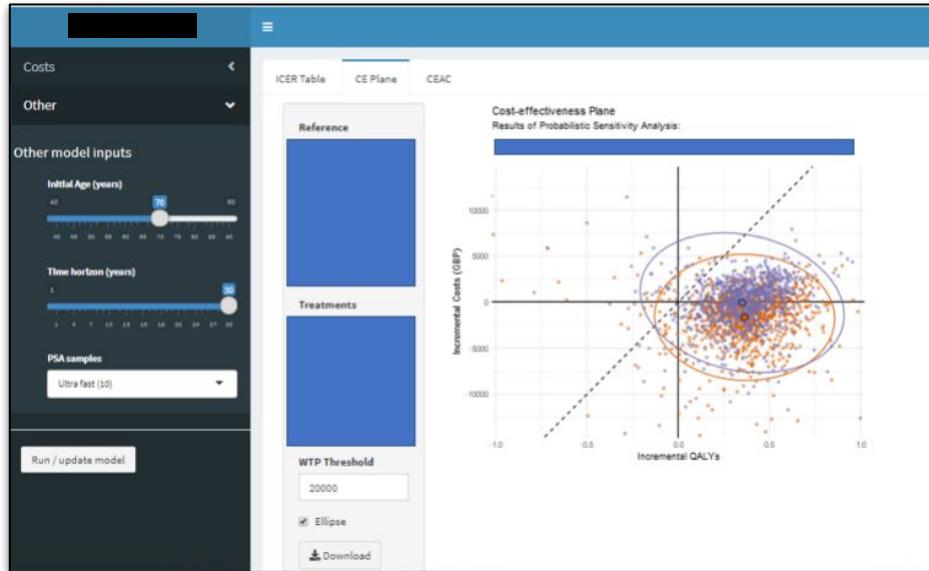


https://darkpeakanalytics.shinyapps.io/living_HTA_demo/



<https://shiny.york.ac.uk/shortfall/>

Making Health Economic Models Shiny



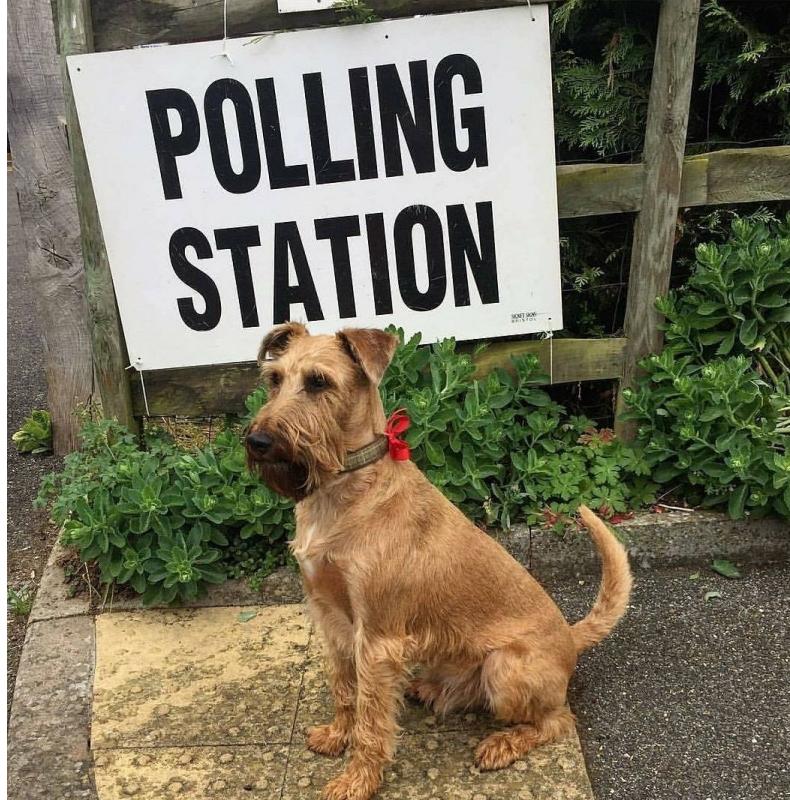
Case Study: shiny user-interface to allow stakeholder to play with the model and investigate effect of using updated evidence in the model. As a result a bug in the model was discovered, and fixed. Because the stakeholder was not an R programmer the bug may not have otherwise been discovered.

Making Health Economic Models Shiny



Shiny is like a superhero costume for R-based health economic models, it gives them a flashy makeover and makes them more presentable to the decision maker. But a fancy app does not substitute for a well constructed model .

Living HTA



Living HTA

PharmacoEconomics (2023) 41:227–237
<https://doi.org/10.1007/s40273-022-01229-4>

LEADING ARTICLE



Living Health Technology Assessment: Issues, Challenges and Opportunities

Praveen Thokala¹ · Tushar Srivastava^{1,2} · Robert Smith^{1,3} · Shijie Ren¹ · Melanie D. Whittington⁴ · Jamie Elvidge⁵ · Ruth Wong¹ · Lesley Uttley¹

Accepted: 4 December 2022 / Published online: 18 January 2023
 © The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

Health technology assessments (HTAs) are typically performed as one-off evaluations and can potentially become out-of-date due to the availability of new data, new comparators, or other factors. Recently, living approaches have been applied to systematic reviews and network meta-analyses to enable evidence syntheses to be updated more easily. In this paper, we provide a definition for ‘Living HTA’ where such a living approach could be applied to the entire HTA process. Living HTA could involve performing regular or scheduled updates using a traditional manual approach, or indeed in a semi-automated manner leveraging recent technological innovations that automate parts of the HTA process. The practical implementation of living HTA using both approaches (i.e., manual approach and using semi-automation) is described along with the likely issues and challenges with planning and implementing a living HTA process. The time, resources and additional considerations outlined may prohibit living HTA from becoming the norm for every evaluation; however, scenarios where living HTA would be particularly beneficial are discussed.

Key Points for Decision Makers

Health technology assessments (HTAs) are typically performed as one-off evaluations and can quickly become out-of-date.

Living HTA approaches can ensure that the HTAs are up-to-date, and potentially living HTAs could be updated manually or (semi-)automatically using innovative software platforms.

However, living HTA involves substantial time, planning and resource commitments, and as such should only be used in situations where it is important to ensure the HTA is up-to-date.

1 Introduction

Health technology assessment (HTA) agencies perform evaluation of clinical and cost-effectiveness evidence of new interventions to decide whether they should be reimbursed.

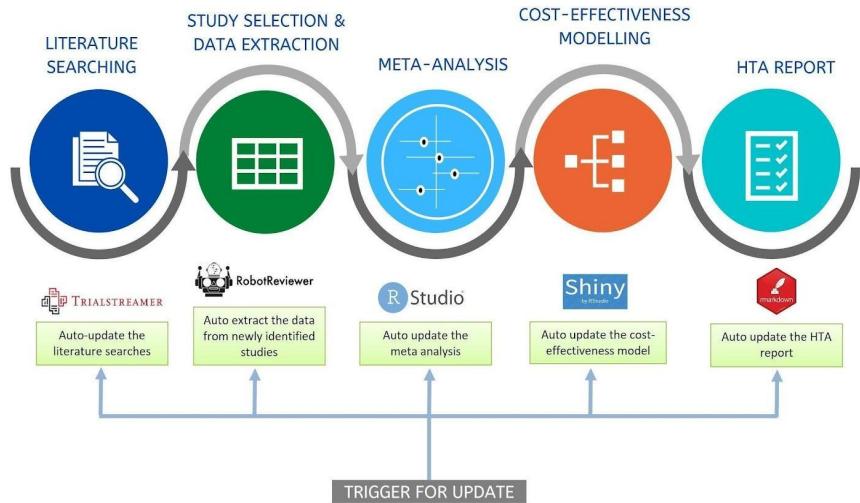
Extended author information available on the last page of the article

These are typically performed as one-off evaluations and can potentially become out-of-date due to various reasons (e.g., availability of new data, new comparators, new methods etc.). While some HTA agencies perform updates of HTAs periodically if certain criteria are met, these updates are typically a few years apart and the results of updates may already be out-of-date by the time of the publication.

There is growing recognition of the need for the HTA process to respond to an evolving evidence base, particularly in reimbursement decisions with high uncertainty. A recent paper on ‘Life-cycle HTA’ suggests that HTA must explore the value of health technologies from inception through maturity, and proposes a model for integrating changes arising from new evidence to feed into adoption, no adoption and disinvestment decisions [1]. However, as far as we know, there is no literature on the practicalities of performing a responsive, dynamic HTA (which we call ‘Living HTA’).

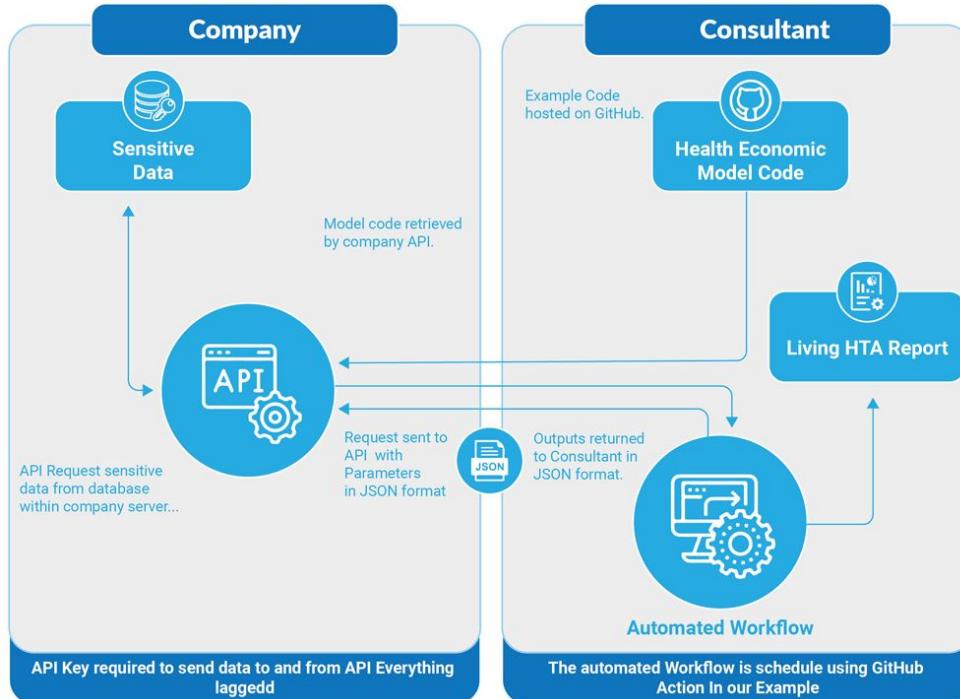
While examples of living systematic reviews and living meta-analyses are already well established, living HTA is not yet fully defined or understood. In this paper, we outline the ways in which the different parts of HTA can become out-of-date, and provide a definition for living HTA and the situations in which it could be useful. While there are similarities between living HTA and the updates that HTA groups make, we outline how the living HTA process could potentially be operationalised from the outset, provide

Figure 3: Potential example living HTA using (semi-)automation



Thokala, P., Srivastava, T., **Smith, R.**, Ren, S., Whittington, M.D., Elvidge, J., Wong, R., Uttley, L. (2023). Living Health Technology Assessment – Issues, Challenges and Opportunities. *Pharmacoeconomics*.

Living HTA



Wellcome Open Research Wellcome Open Research 2022, 7:194 Last updated: 24 OCT 2022

Check for updates

METHOD ARTICLE

Living HTA: Automating Health Economic Evaluation with R [version 2; peer review: 2 approved]

Robert A. Smith  1,3, Paul P. Schneider  1,3, Wael Mohammed  1,3

¹School of Health and Related Research, University of Sheffield, Sheffield, S1 4DA, UK
²Community, Sheffield, S1 2GD, UK
³Dark Peak Analytics, Sheffield, S11 7BA, UK

V2 First published: 21 Jul 2022, 7:194
<https://doi.org/10.26968/wellcomeresearch.v7.194>
Latest published: 11 Oct 2022, 7:194
<https://doi.org/10.26968/wellcomeresearch.v7.194>

Open Peer Review

Approval Status	1	2
version 2		
(revision)		
version 1		
(revision)		

1. Mohsen Sadatrafai  University of British Columbia, Vancouver, Canada
2. Devin Ingersoll  EntityRisk Inc., Princeton, USA

Any reports and responses or comments on the article can be found at the end of the article.

Abstract
Background: Requiring access to sensitive data can be a significant obstacle for the development of health models in the Health Economics & Outcomes Research (HEOR) setting. We demonstrate how health economic evaluation can be conducted with minimal transfer of data between parties, while automating reporting as new information becomes available.

Methods: We developed an automated analysis and reporting pipeline for health economic evaluations that can be made available openly on a GitHub repository. The pipeline consists of three parts: An economic model is constructed by the consultant using pseudo data. On the data-owner side, an application programming interface (API) is hosted on a server. This API hosts all sensitive data, so that it need not be provided to the consultant. An automated workflow is created, which calls the API, retrieves results, and generates a report.

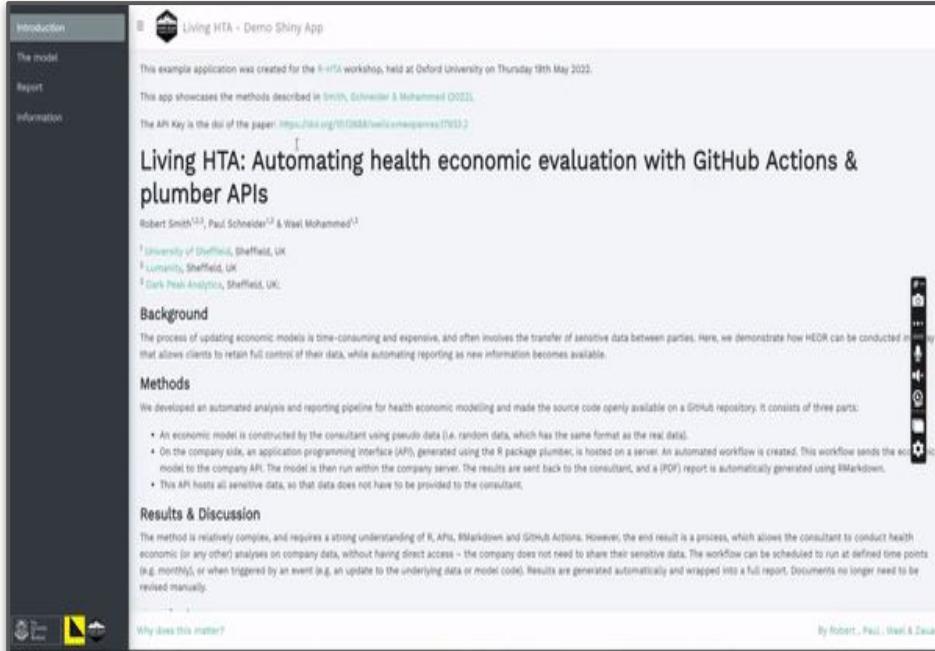
Results: The application of modern data science tools and practices allows analyses of data without the need for direct access, negating the need to transfer data. In addition, the entire workflow can be largely automated; the analysis can be scheduled to run at defined time points (e.g. monthly), or when triggered by an event (e.g. an update to the underlying data). The results can be generated automatically and then be reported in a report. Documents no longer need to be revised manually.

Conclusions: This example demonstrates that it is possible, within a HEOR setting, to separate the health economic model from the data, and automate the main steps of the analysis pipeline.

Keywords
HEOR, HTA, APIs, R, plumber

Page 1 of 27

Living HTA



The screenshot shows a shiny application window titled "Living HTA - Demo Shiny App". The left sidebar contains navigation links: "Introduction", "The model", "Report", and "Information". The main content area displays the following text:

This example application was created for the R-HTA workshop, held at Oxford University on Thursday 19th May 2022.

This app showcases the methods described in Smith, Schneider & Mohammed (2022).

The API Key is the doi of the paper: <https://doi.org/10.1388/weconomics.17933.2>

Living HTA: Automating health economic evaluation with GitHub Actions & plumber APIs

Robert Smith^{1,2}, Paul Schneider^{1,2} & Wael Mohammed^{1,2}

¹University of Sheffield, Sheffield, UK
²Commons, Sheffield, UK
²Dark Peak Analytics, Sheffield, UK

Background

The process of updating economic models is time-consuming and expensive, and often involves the transfer of sensitive data between parties. Here, we demonstrate how HEDR can be conducted in a way that allows clients to retain full control of their data, while automating reporting as new information becomes available.

Methods

We developed an automated analysis and reporting pipeline for health economic modeling and made the source code openly available on a GitHub repository. It consists of three parts:

- An economic model is constructed by the consultant using pseudo data (i.e. random data, which has the same format as the real data).
- On the company side, an application programming interface (API), generated using the R package `plumber`, is hosted on a server. An automated workflow is created. This workflow sends the economic model to the company API. The model is then run within the company server. The results are sent back to the consultant, and a (PDF) report is automatically generated using `Markdown`.
- This API heats all sensitive data, so that data does not have to be provided to the consultant.

Results & Discussion

The method is relatively complex, and requires a strong understanding of R, APIs, `Markdown` and GitHub Actions. However, the end result is a process, which allows the consultant to conduct health economic (or any other) analyses on company data, without having direct access – the company does not need to share their sensitive data. The workflow can be scheduled to run at defined time points (e.g. monthly), or when triggered by an event (e.g. an update to the underlying data or model code). Results are generated automatically and wrapped into a full report. Documents no longer need to be revised manually.

Why does this matter?

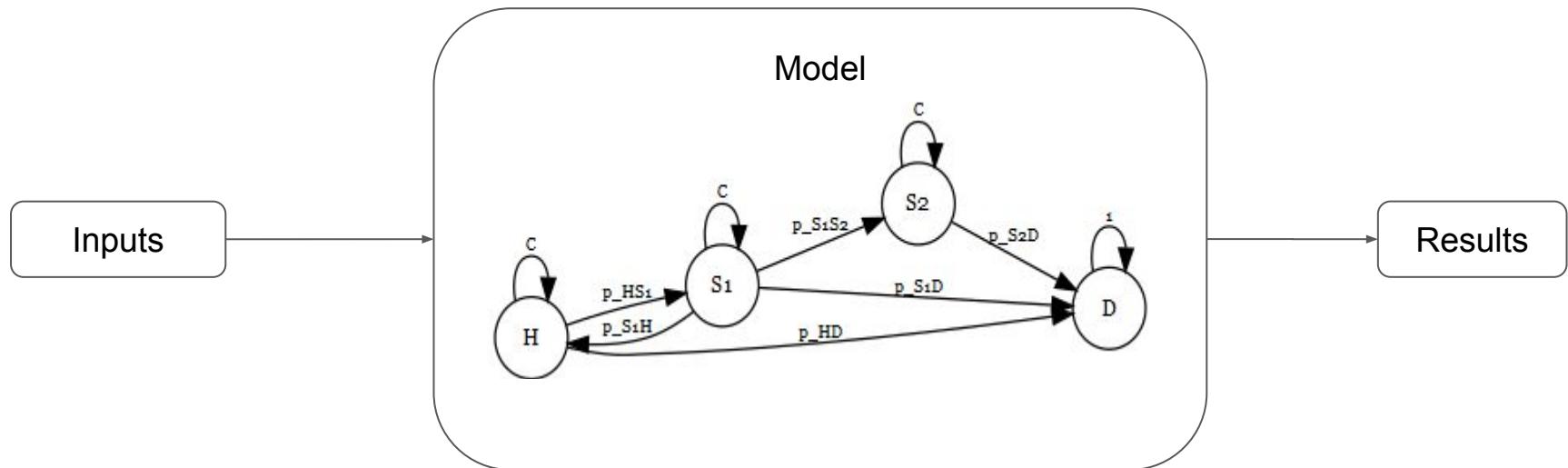
By Robert, Paul & Wael

https://darkpeakanalytics.shinyapps.io/living_HTA_demo/

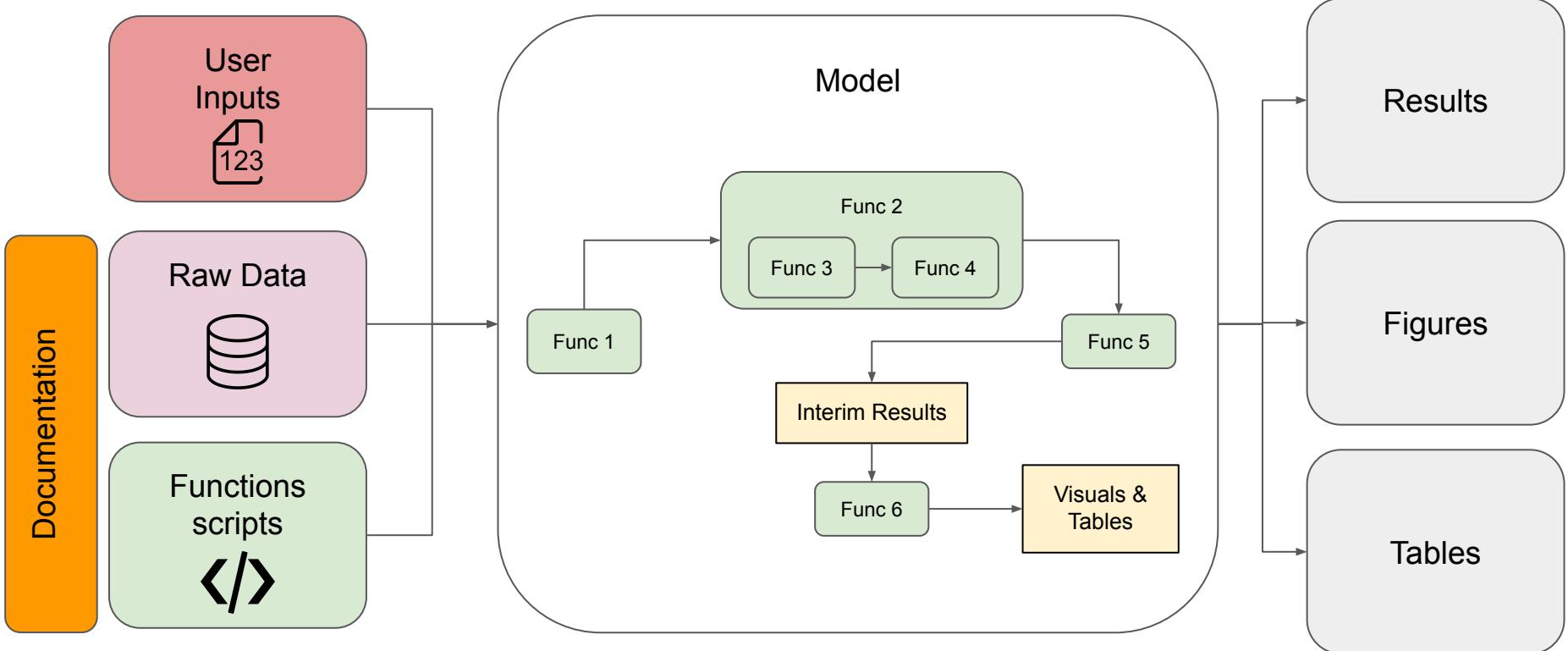
Packaging Cost-effectiveness models



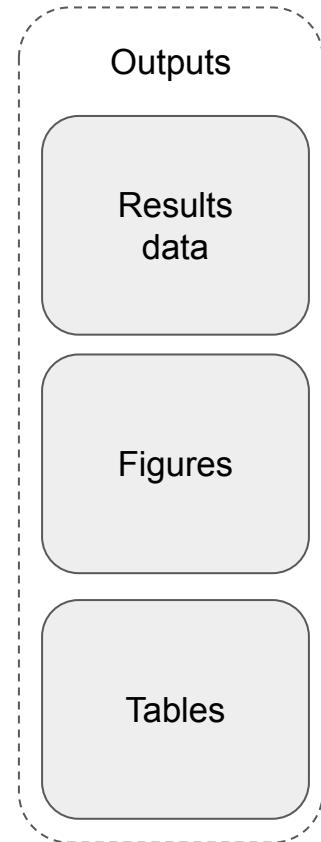
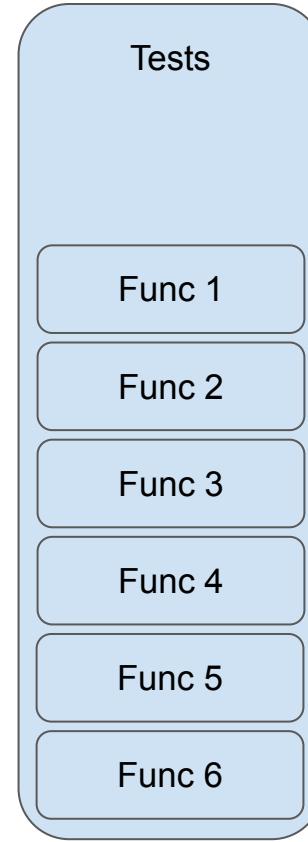
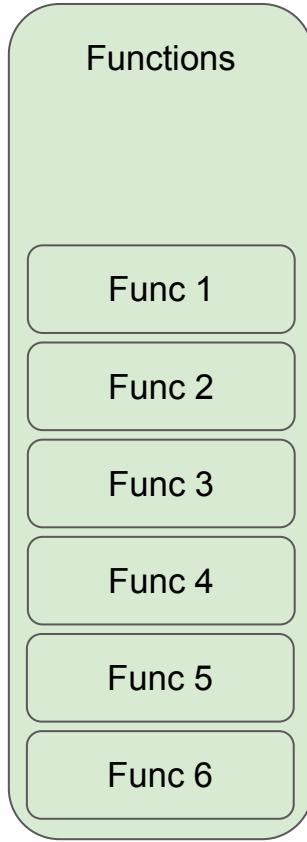
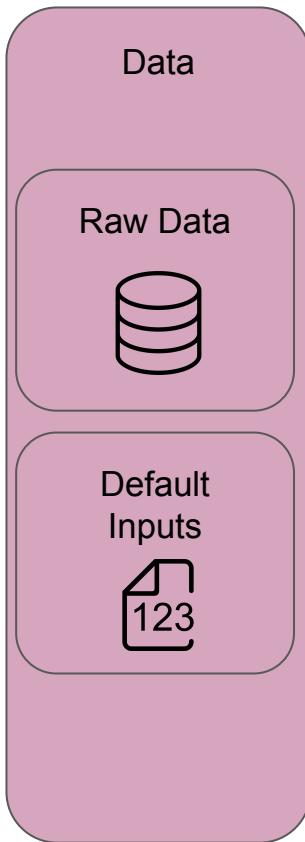
Building a model in R



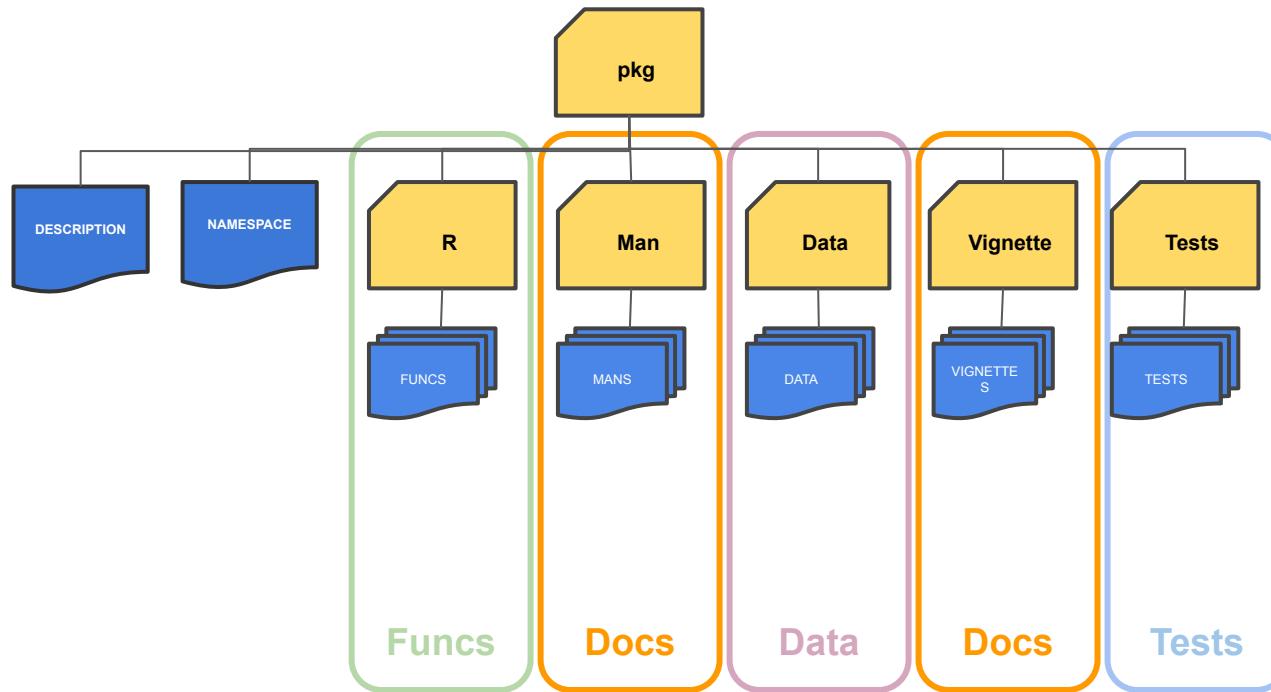
Building a model in R



Building a model in R



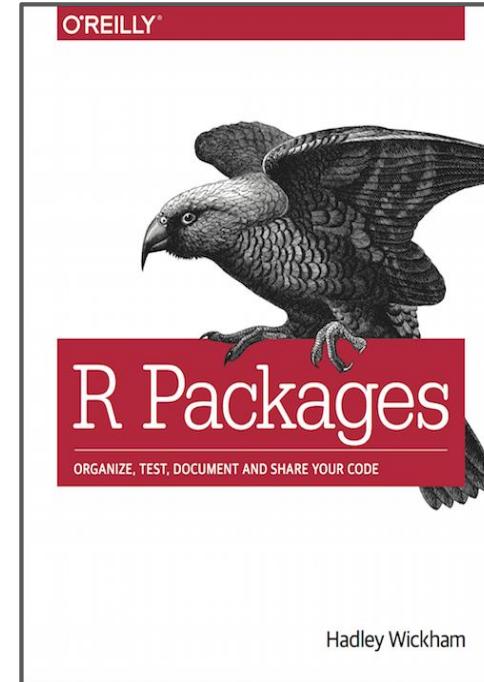
Building a package in R



Benefits of using a Package vs non-packaged code.

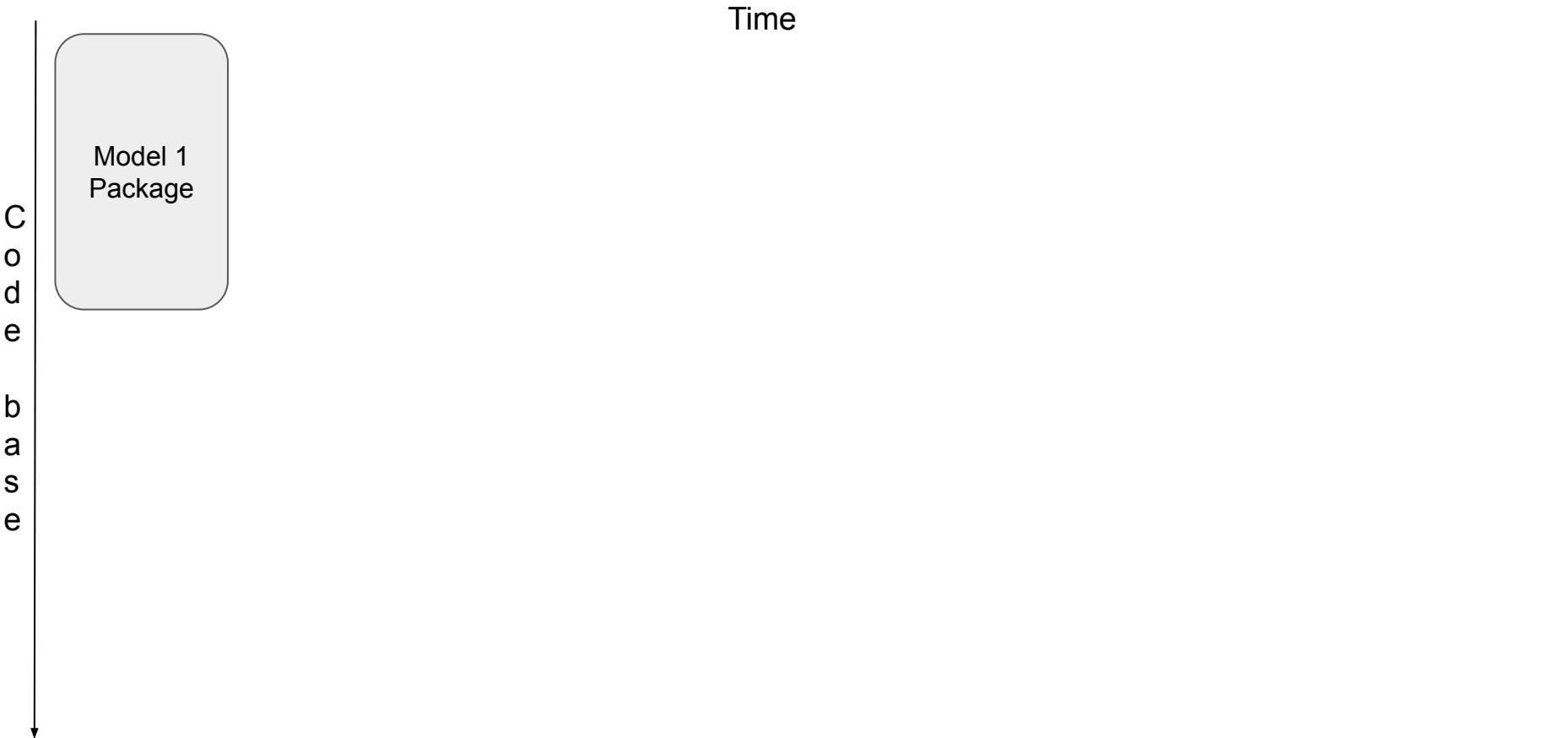


1. Every Package will have a similar structure
 - a. Improves familiarity with models.
2. Documentation by default
 - a. Vignettes to show how the package works (walking the user through the code).
 - b. Roxygen comments on every function (exactly what is it doing)
3. Unit testing is built-in
 - a. Testing gives modeller confidence in their methods.
 - b. Testing allows reviewers to 'test the tests' rather than from scratch.
4. Functions are more easily distributed (e.g.
`install_github("your-package")`)
 - a. Therefore don't have to continually re-invent wheels
 - b. Standardisation (pros and cons)
 - c. Validation (easier to review, more confidence)



<https://r-pkgs.org/>

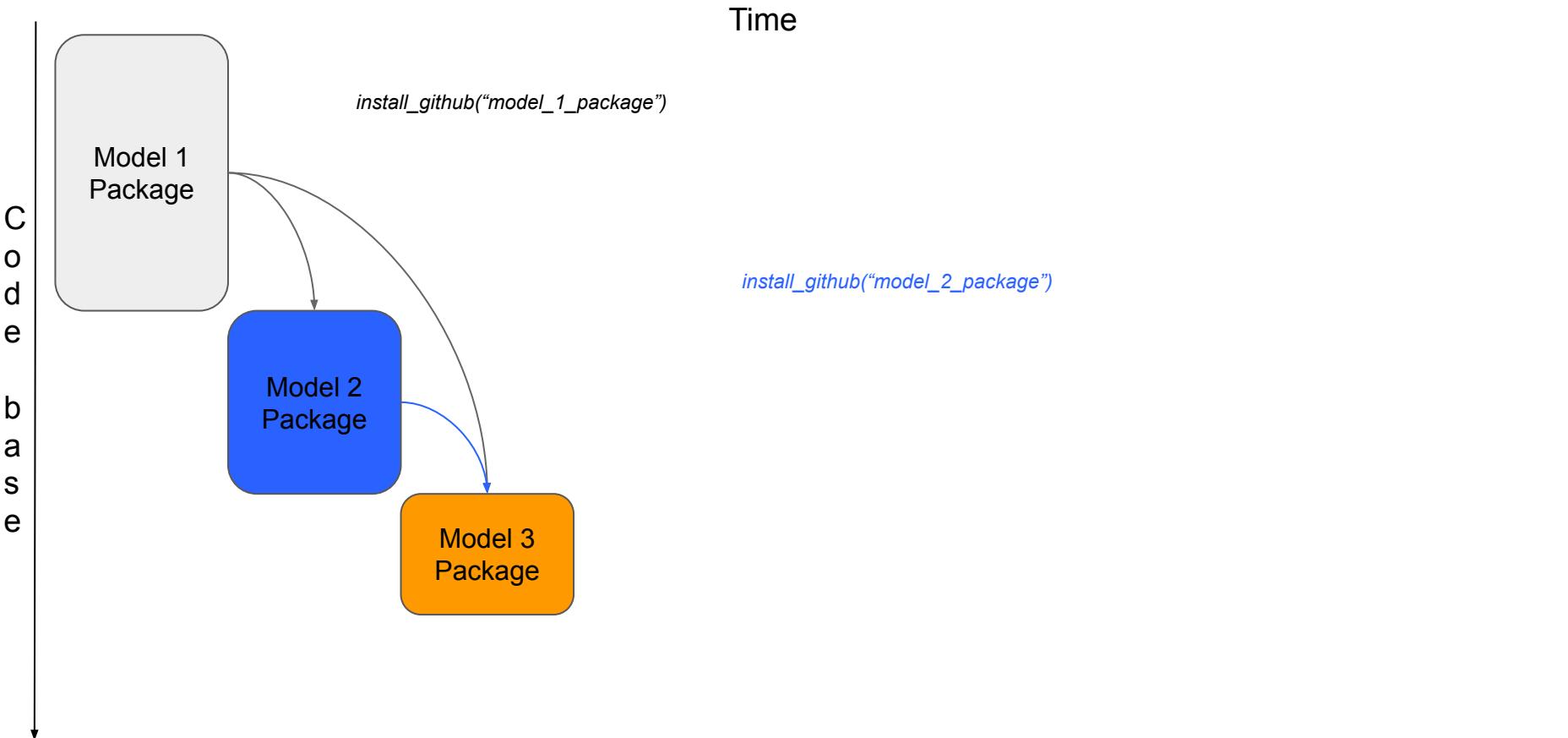
Benefits of using a Package vs non-packaged code.



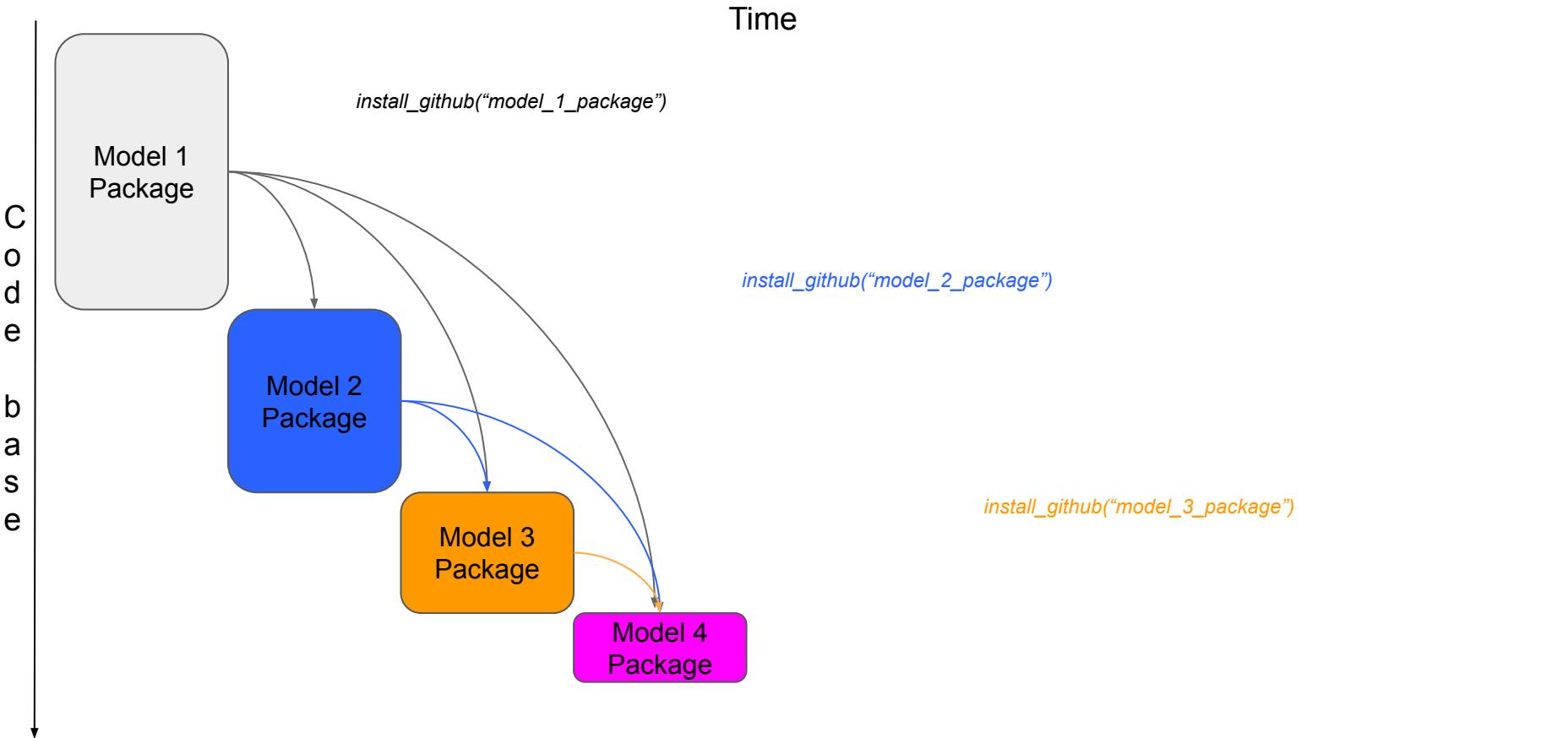
Benefits of using a Package vs non-packaged code.



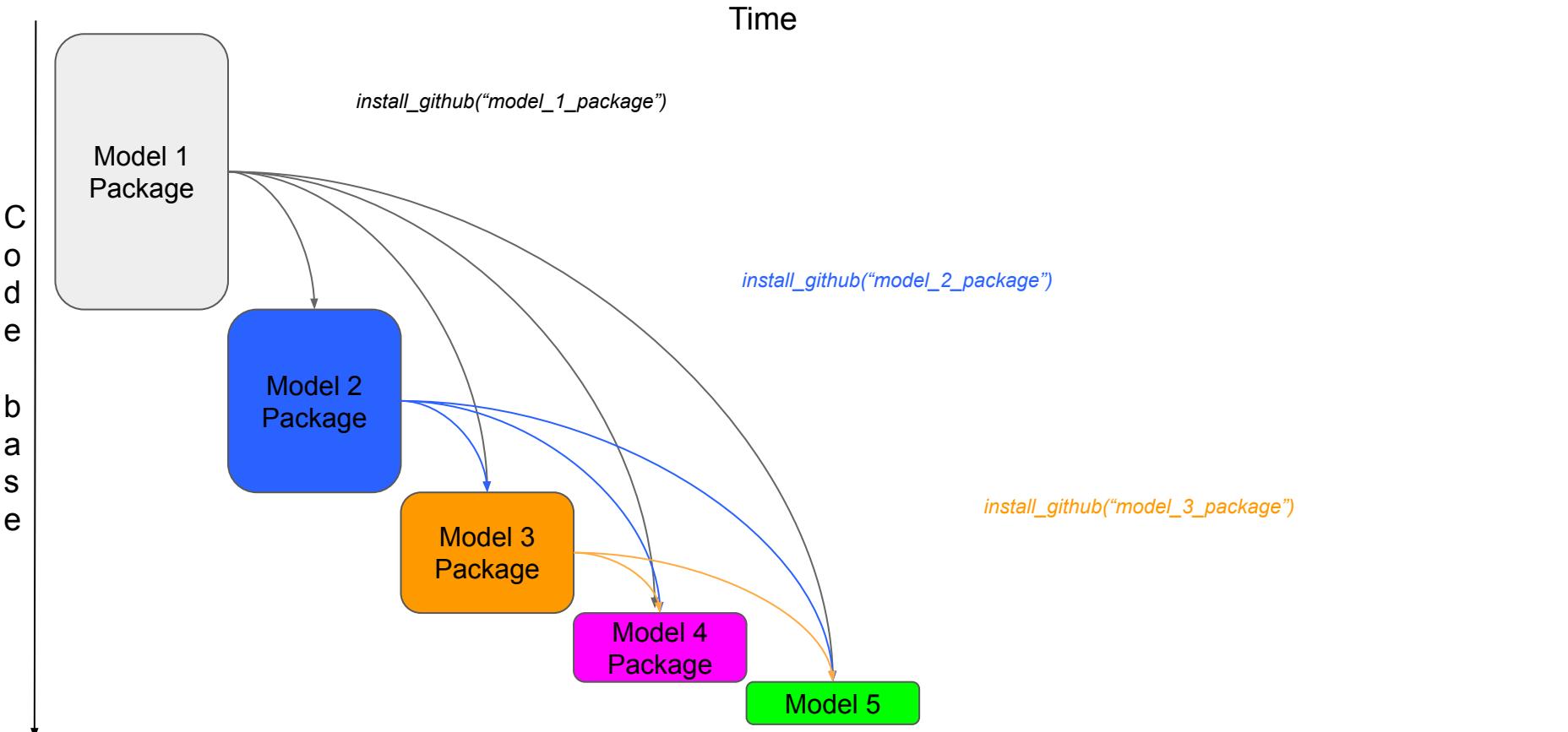
Benefits of using a Package vs non-packaged code.



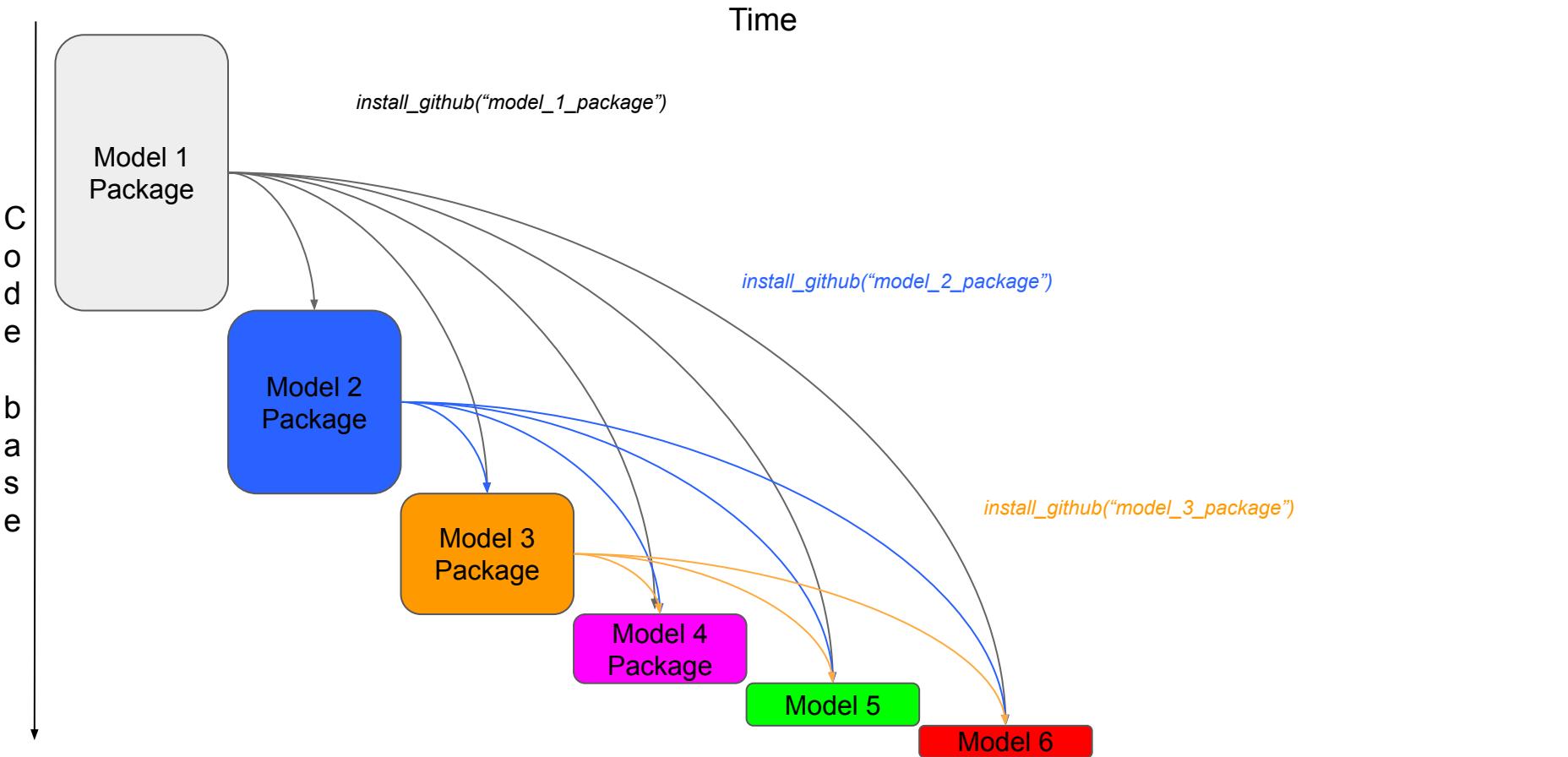
Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



*“But won’t we end up having to install
a large number of packages, just to
get one or two functions from each?”*

Anon



Large package, small function ...

Benefits of using a Package vs non-packaged code.



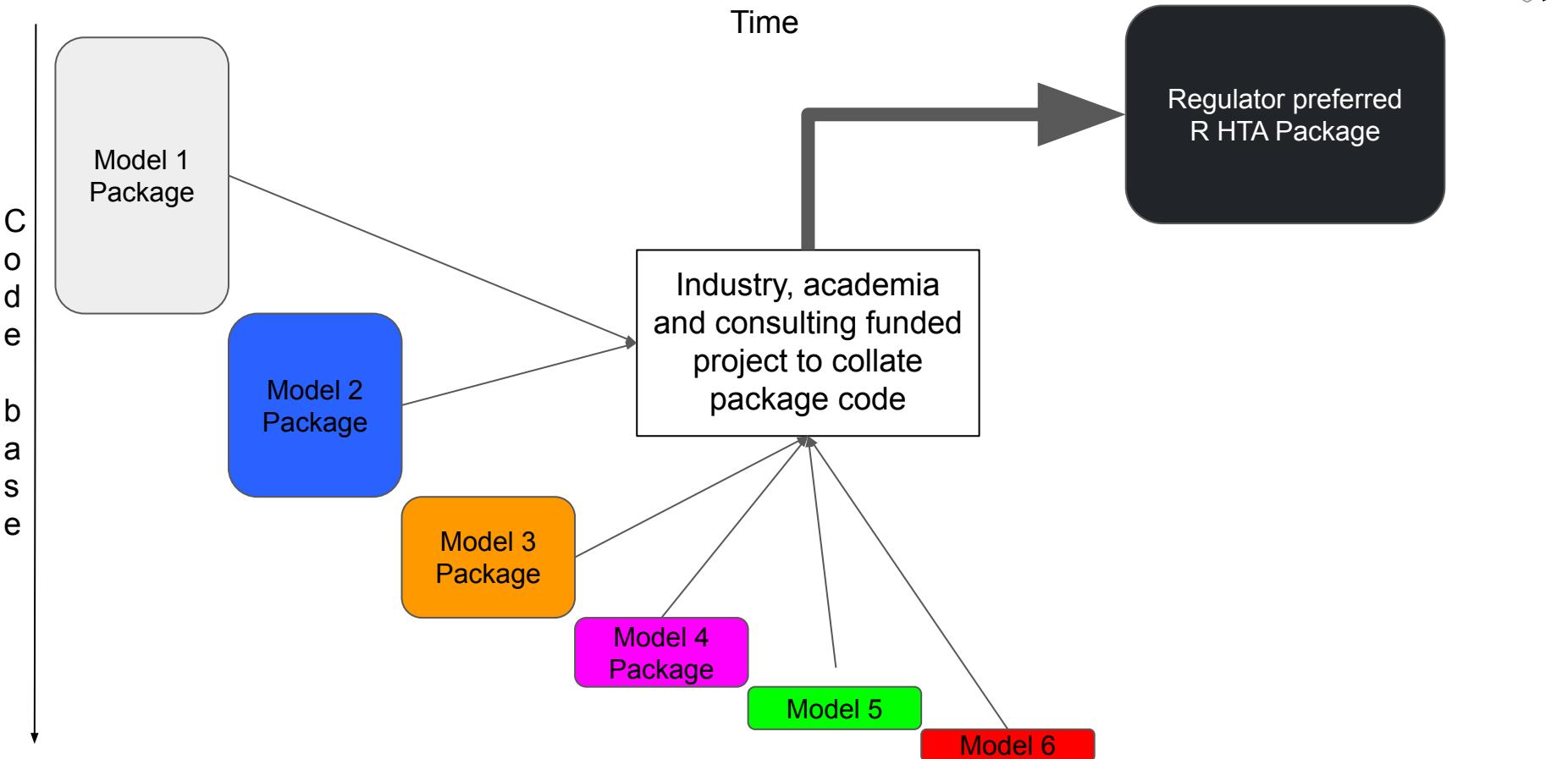
*“But won’t we end up having to install
a large number of packages, just to
get one or two functions from each?”*

Anon (actually it was Paul)

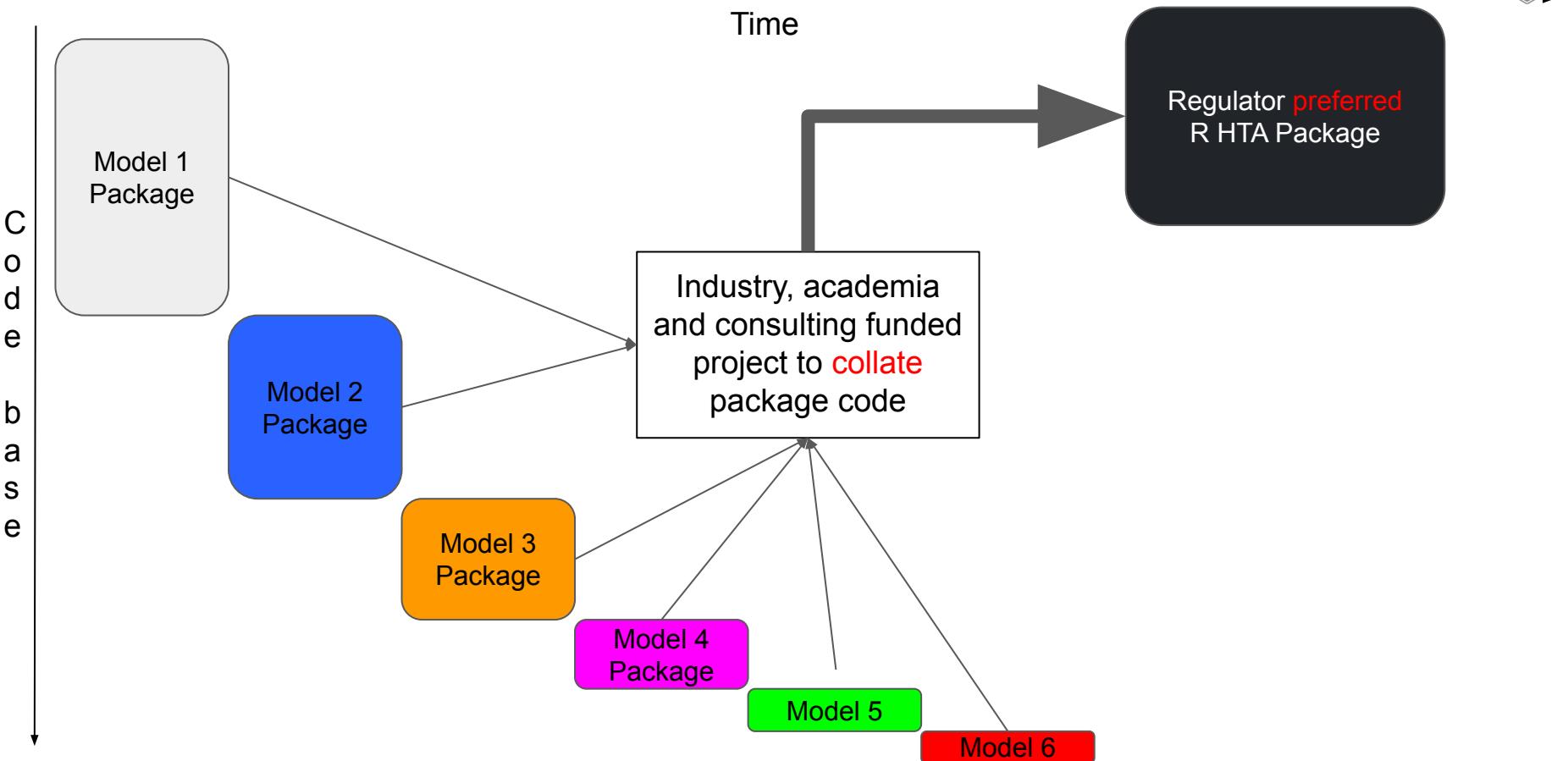


Large package, small function ...

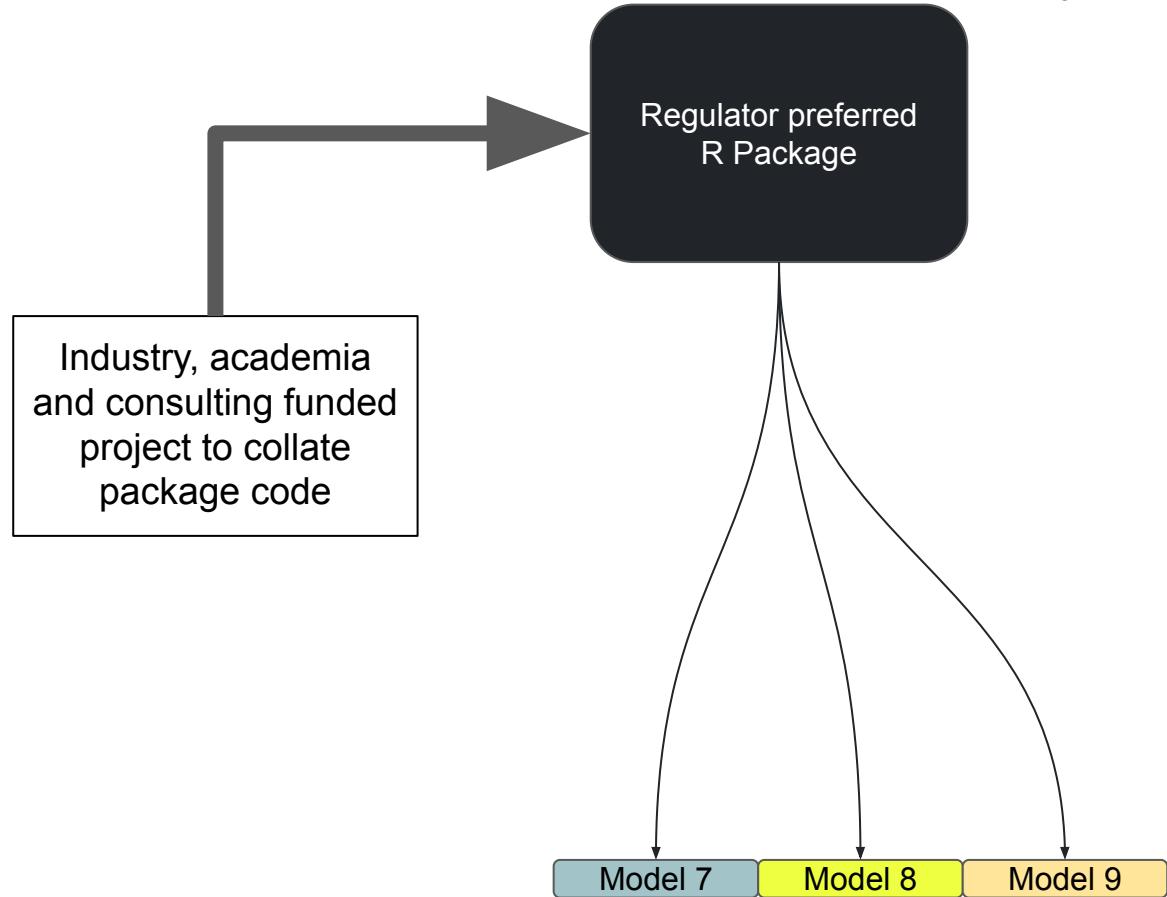
Benefits of using a Package vs non-packaged code.



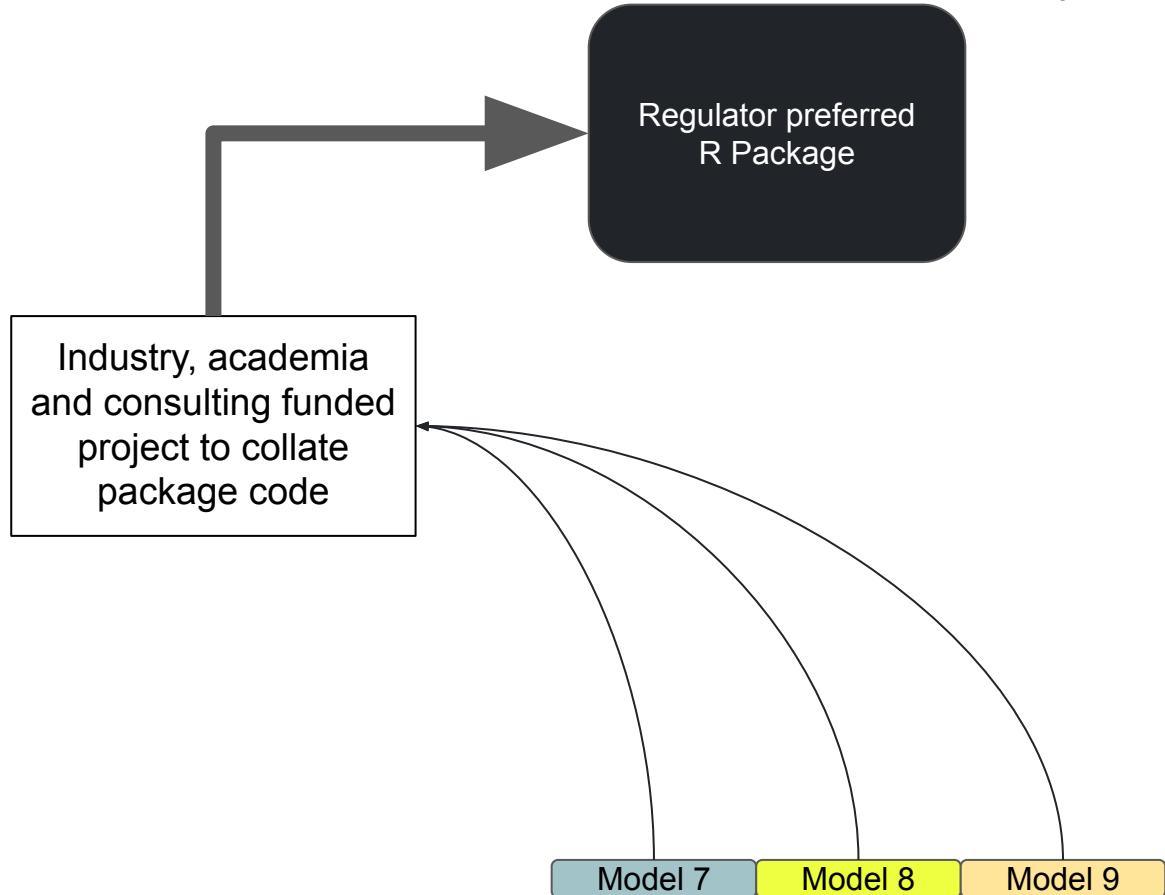
Benefits of using a Package vs non-packaged code.



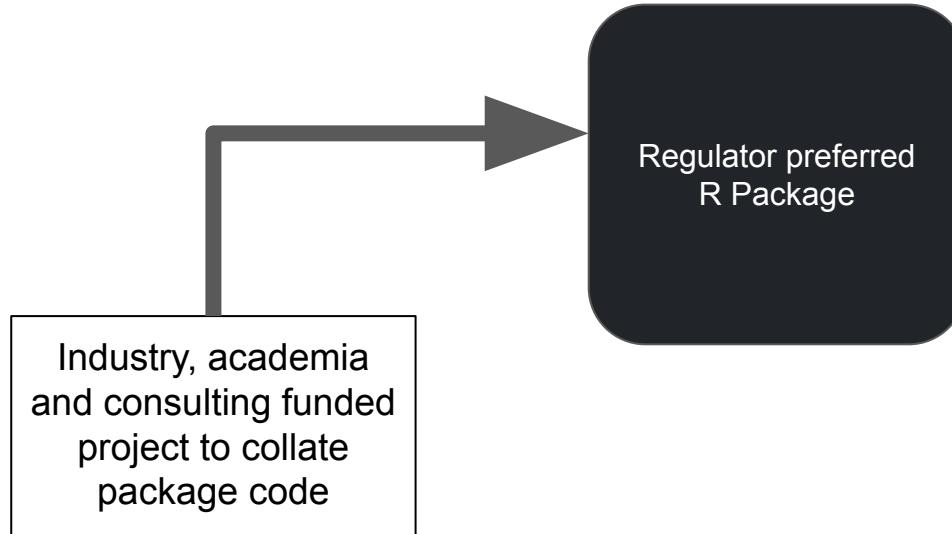
Benefits of using a Package vs non-packaged code.



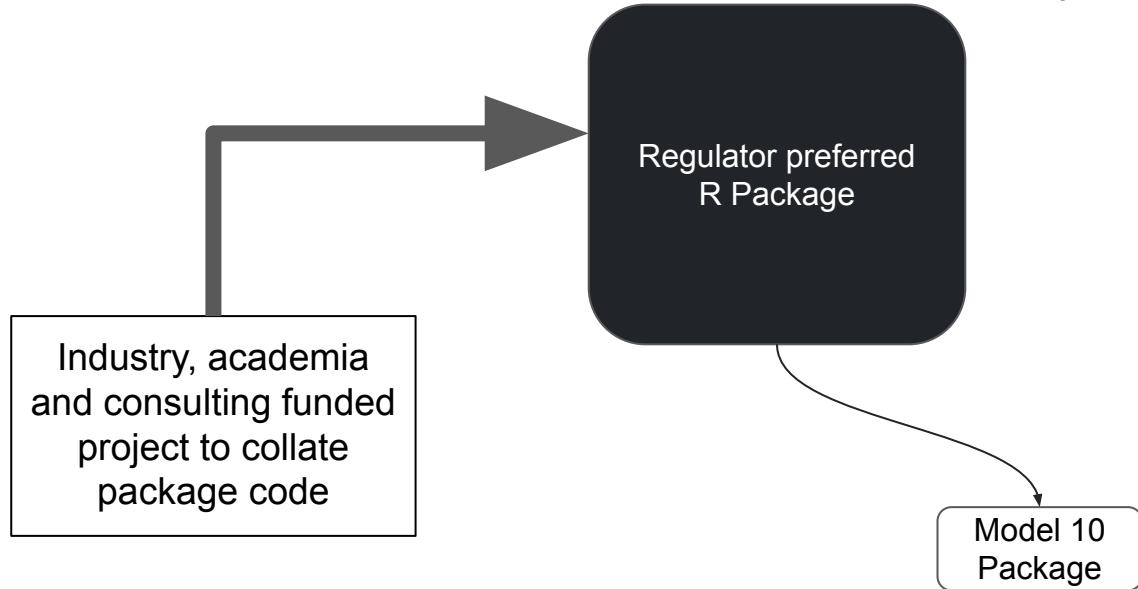
Benefits of using a Package vs non-packaged code.



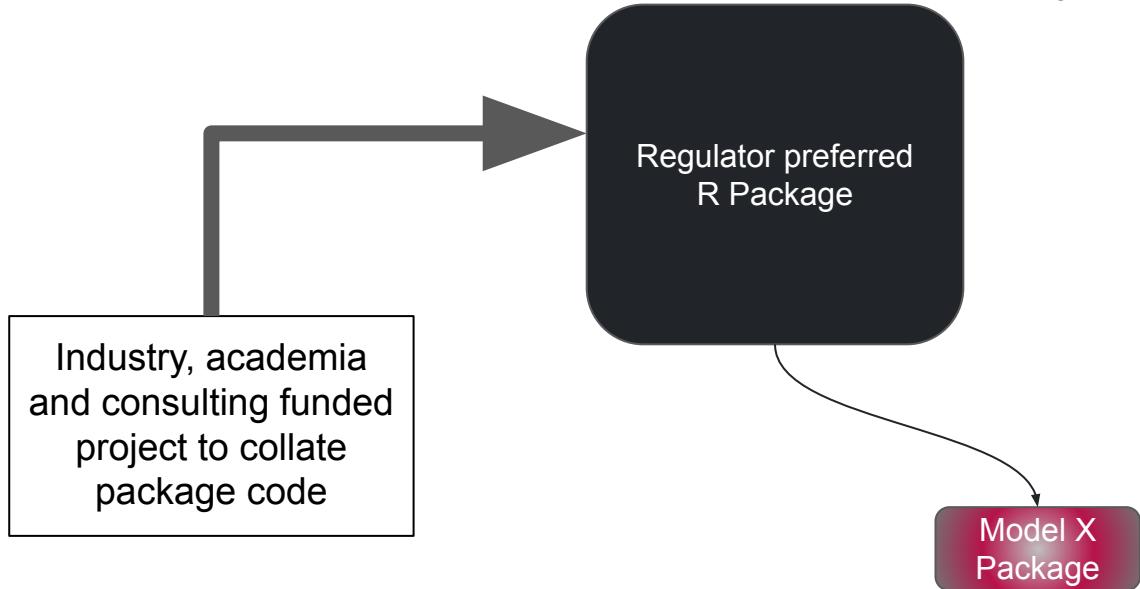
Benefits of using a Package vs non-packaged code.



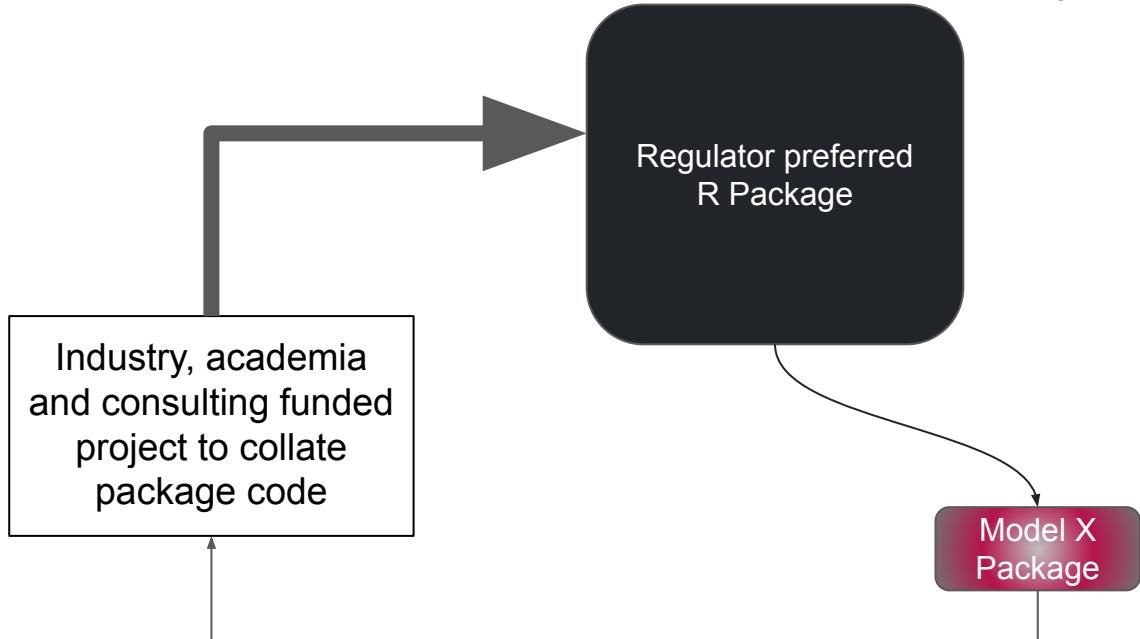
Benefits of using a Package vs non-packaged code.



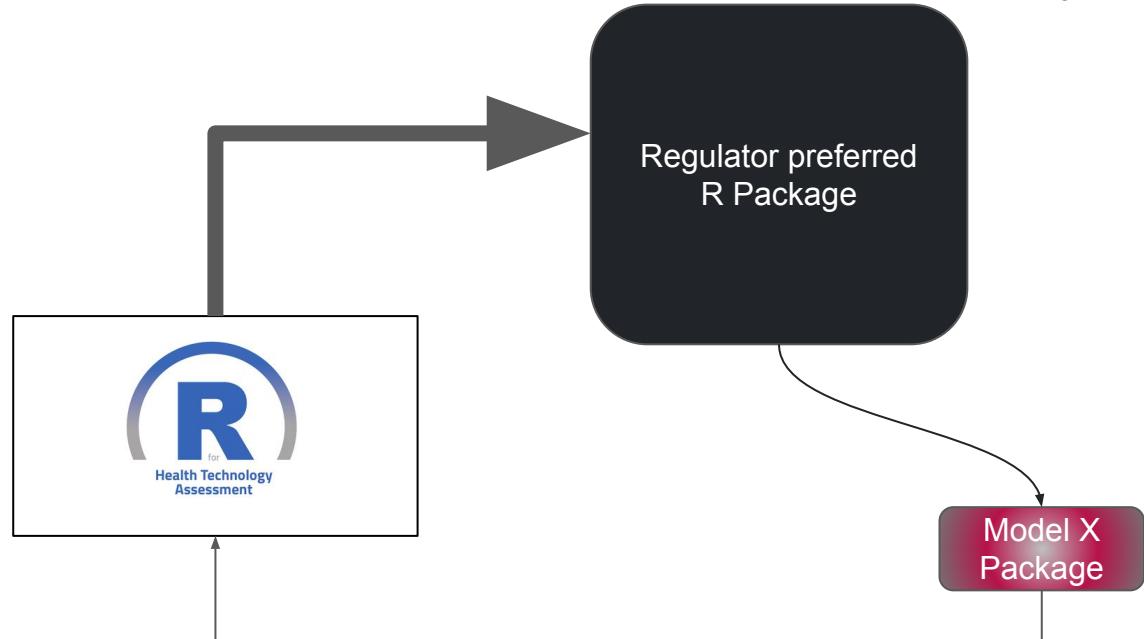
Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



Benefits of using a Package vs non-packaged code.



- R Packages can serve as **templates for best-practice** in health economic model building.
- They make it **easier to review** since code is documented and unit tested by default.
- They make it **easier to distribute code** so that others can apply the same methods.
- **Confidence is crucial**, there is a key role for trusted experts to give legitimacy to packages.
- This role would be substantial and ongoing indefinitely ... but ...
- It would result in a huge **gain in efficiency and quality** of health economic models.
- For individual researchers, attribution would be beneficial for their profile.

Previous examples



Economic Evaluation

CDX2 Biomarker Testing and Adjuvant Therapy for Stage II Colon Cancer: An Exploratory Cost-Effectiveness Analysis

Fernando Alarid-Escudero, PhD, Deborah Schrag, MD, MPH, Karen M. Kuntz, ScD

ABSTRACT

Objectives: Adjuvant chemotherapy is not recommended for patients with average-risk stage II (T3N0) colon cancer. Nevertheless, a subgroup of these patients who are CDX2-negative might benefit from adjuvant chemotherapy. We evaluated the cost-effectiveness of testing for the absence of CDX2 expression followed by adjuvant chemotherapy (fluorouracil combined with oxaliplatin [FOLFOX]) for patients with stage II colon cancer.

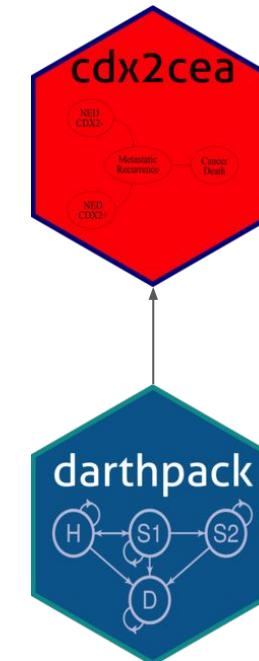
Methods: We developed a decision model to simulate a hypothetical cohort of 65-year-old patients with average-risk stage II colon cancer with 7.2% of these patients being CDX2-negative under 2 different interventions: (1) test for the absence of CDX2 expression followed by adjuvant chemotherapy for CDX2-negative patients and (2) no CDX2 testing and no adjuvant chemotherapy for any patient. We derived disease progression parameters, adjuvant chemotherapy effectiveness and utilities from published analyses, and cancer care costs from the Surveillance, Epidemiology, and End Results (SEER)-Medicare data. Sensitivity analyses were conducted.

Results: Testing for CDX2 followed by FOLFOX for CDX2-negative patients had an incremental cost-effectiveness ratio of \$5500/quality-adjusted life-years (QALYs) compared with no CDX2 testing and no FOLFOX (6.874 vs 6.838 discounted QALYs and \$89,991 vs \$89,797 discounted US dollar lifetime costs). In sensitivity analyses, considering a cost-effectiveness threshold of \$100,000/QALY, testing for CDX2 followed by FOLFOX for CDX2-negative patients remains cost-effective for hazard ratios of <0.975 of the effectiveness of FOLFOX in CDX2-negative patients in reducing the rate of developing a metastatic recurrence.

Conclusions: Testing tumors of patients with stage II colon cancer for CDX2 and administration of adjuvant treatment to the subgroup found CDX2-negative is a cost-effective and high-value management strategy across a broad range of plausible assumptions.

Keywords: CDX2, cost-effectiveness analysis, decision-analytic model, immunohistochemistry testing, stage II colon cancer.

VALUE IN HEALTH. 2022; 25(3):409-418

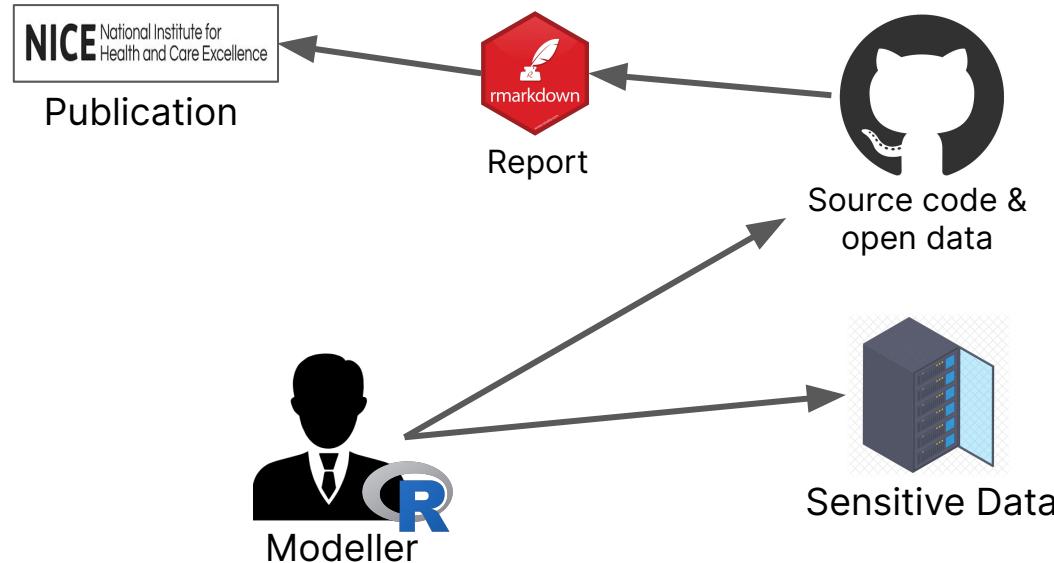


Alarid-Escudero, F., Schrag, D. and Kuntz, K.M., 2022. CDX2 biomarker testing and adjuvant therapy for stage II colon cancer: an exploratory cost-effectiveness analysis. *Value in Health*, 25(3), pp.409-418.

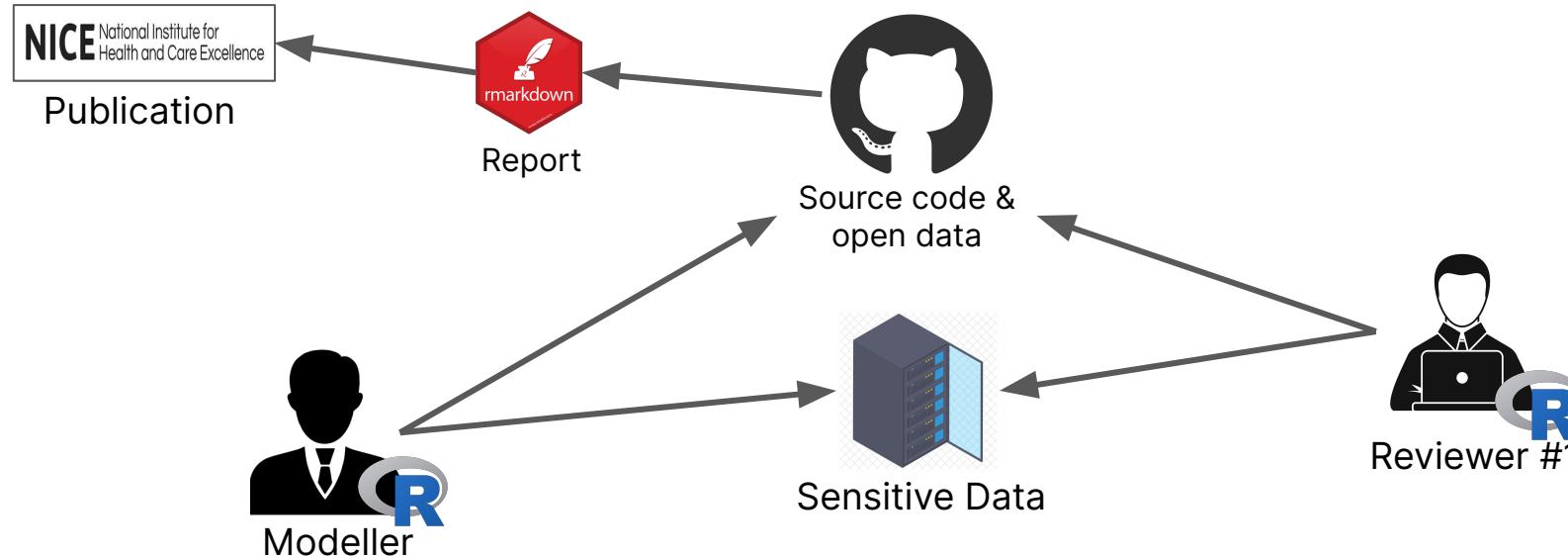
Reviewing Health Economic Models in R



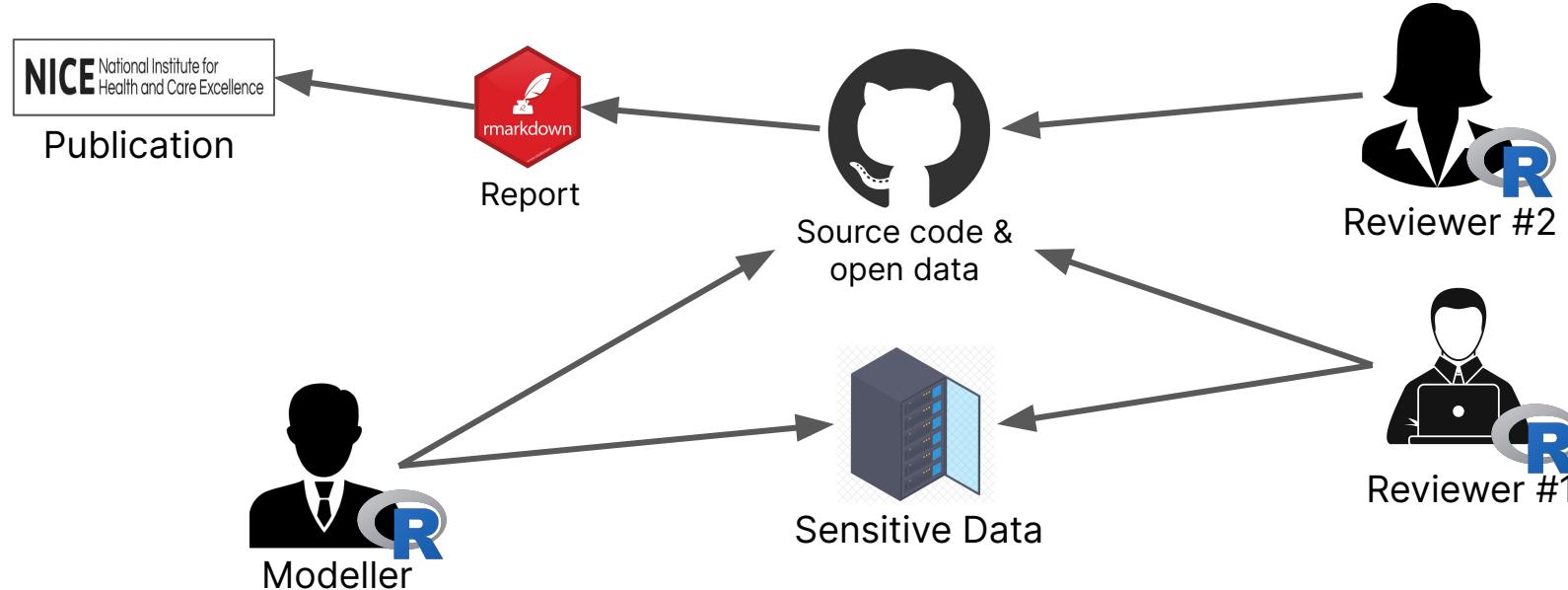
External review options



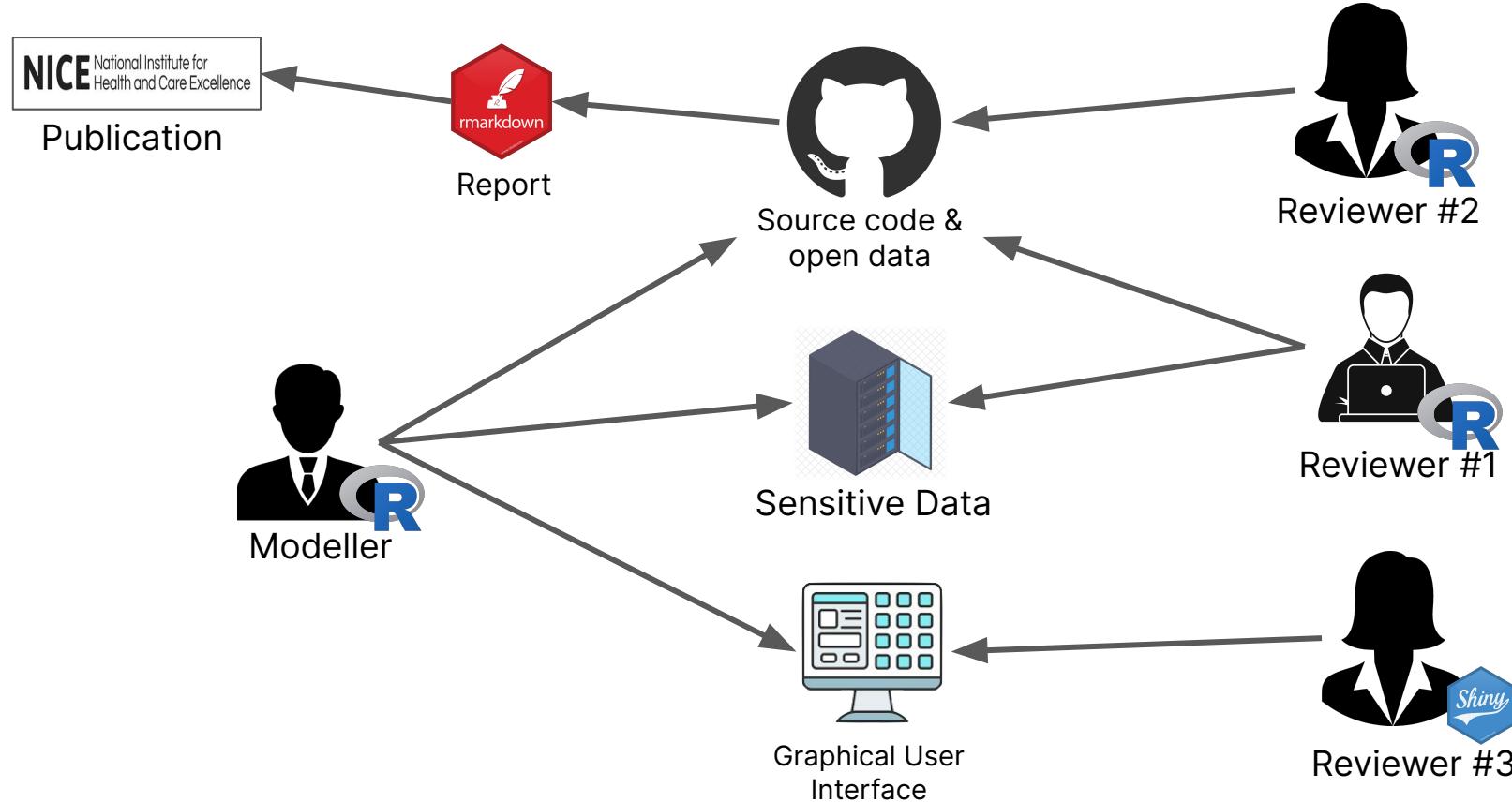
External review options



External review options



External review options



assertHE R package

Aim: *to help modellers build and review health economic models in R.*

Functionality:

- 1. Check that the objects created in models conform to standard rules (e.g. probabilities between 0 and 1).
- 2. Summarise & visualise the structure of a model
 - Plot function network color coded by test coverage.
 - Click on the nodes to see function and test source code and test coverage.
 - Display a LLM generated summary of any function.



<https://github.com/dark-peak-analytics/assertHE>

Using the assertHE R package

```
# install.packages("devtools")
devtools::install_github("dark-peak-analytics/assertHE")

library(assertHE)
```

B: insert checks for common errors into the code

```
check_trans_prob_array(a_P = a_P,
                      stop_if_not = T)
```

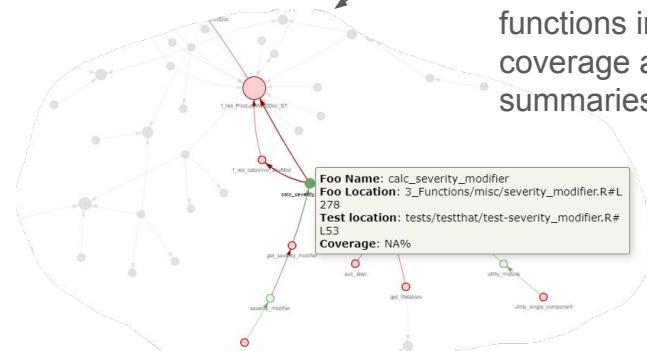
Flags if there are errors or potential problems.

```
# Warning message:
# In check_array_rows_balanced(a_P, stop_if_not = stop_if_not) :
#   Not valid transition probabilities
#   Transition probabilities not valid from Health States:
# 1 H; at cycle 1
# 2 H; at cycle 2
# 3 H; at cycle 3
# 4 H; at cycle 4
# 5 H; at cycle 5
# 6 H; at cycle 6
# 7 H; at cycle 7
# 8 H; at cycle 8
# 9 H; at cycle 9
# 10 H; at cycle 10
```

A: visualise network of functions

```
visualise_project(
  project_path = "path_to_project_directory",
  foo_path = "R",
  test_path = "tests/testthat",
  run_coverage = T)
```

Inspect the network to understand how the model functions interact, their test coverage and get AI function summaries.





assertHE checks

check_trans_prob_array() Checks transition probability array for common errors ensuring the same number of rows and columns, only numerical values between 0 and 1, with rows summing to 1. The function provides confirmation or warnings/error messages as appropriate.

check_markov_trace() Validates Markov trace for feasibility, confirming numerical values, equal row sums, and optionally monotonic increase in dead state proportions, generating confirmation or warnings/error messages.

check_init() Checks initial health state proportions vector for validity, ensuring values within valid probability range, no missing values, sum equals 1, and distinct, non-duplicated names, generating messages for inconsistencies.

plot_PSA_stability() Generates informative plots to inspect stability of results across model iterations, calculating metrics like INMB, ICER, incremental costs, or effects, allowing customization for specific needs, enhancing visual clarity.

assertHE checks

check_trans_prob_array() Checks transition probability array for common errors ensuring the same number of rows and columns, only numerical values between 0 and 1, with rows summing to 1. The function provides confirmation or warnings/error messages as appropriate.

check_markov_trace() Validates Markov trace for feasibility, confirming numerical values, equal row sums, and optionally monotonic increase in dead state proportions, generating confirmation or warnings/error messages.

check_init() Checks initial health state proportions vector for validity, ensuring values within valid probability range, no missing values, sum equals 1, and distinct, non-duplicated names, generating messages for inconsistencies.

plot_PSA_stability() Generates informative plots to inspect stability of results across model iterations, calculating metrics like INMB, ICER, incremental costs, or effects, allowing customization for specific needs, enhancing visual clarity.

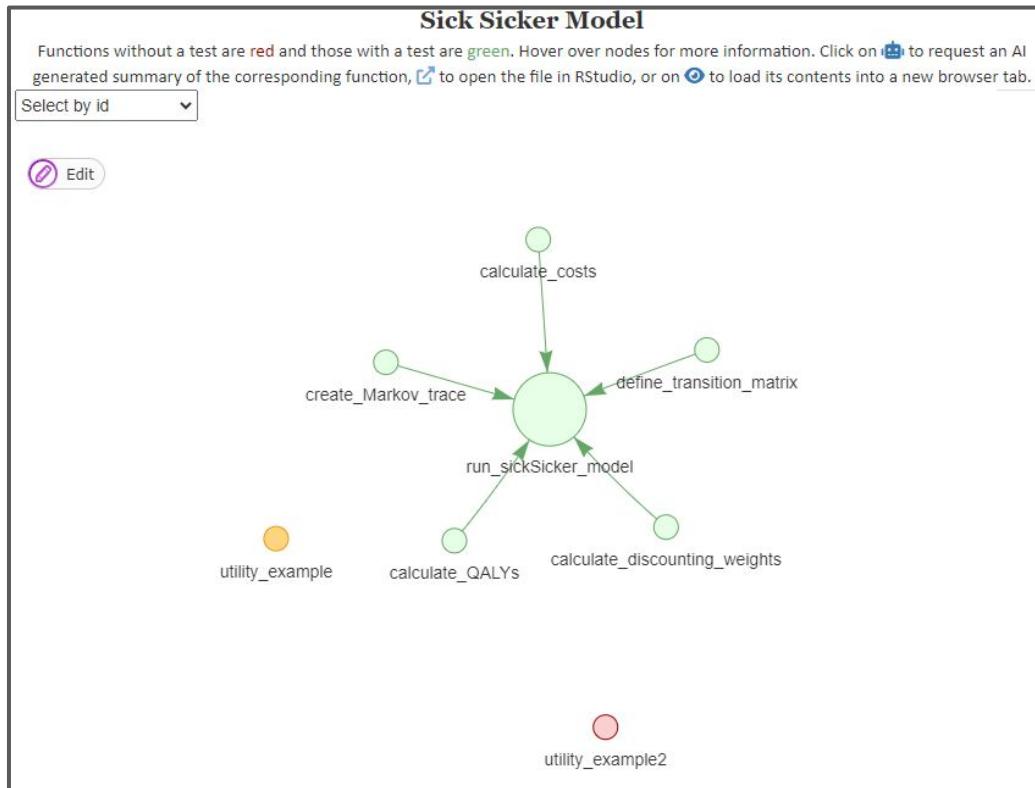
I'm

so

BORED!



assertHE model reviewer



assertHE model reviewer

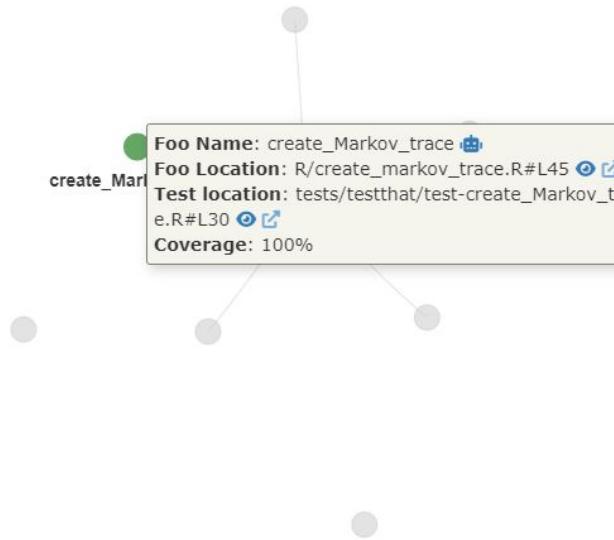
Sick Sicker Model

Functions without a test are **red** and those with a test are **green**. Hover over nodes for more information. Click on  to request an AI generated summary of the corresponding function,  to open the file in RStudio, or on  to load its contents into a new browser tab.

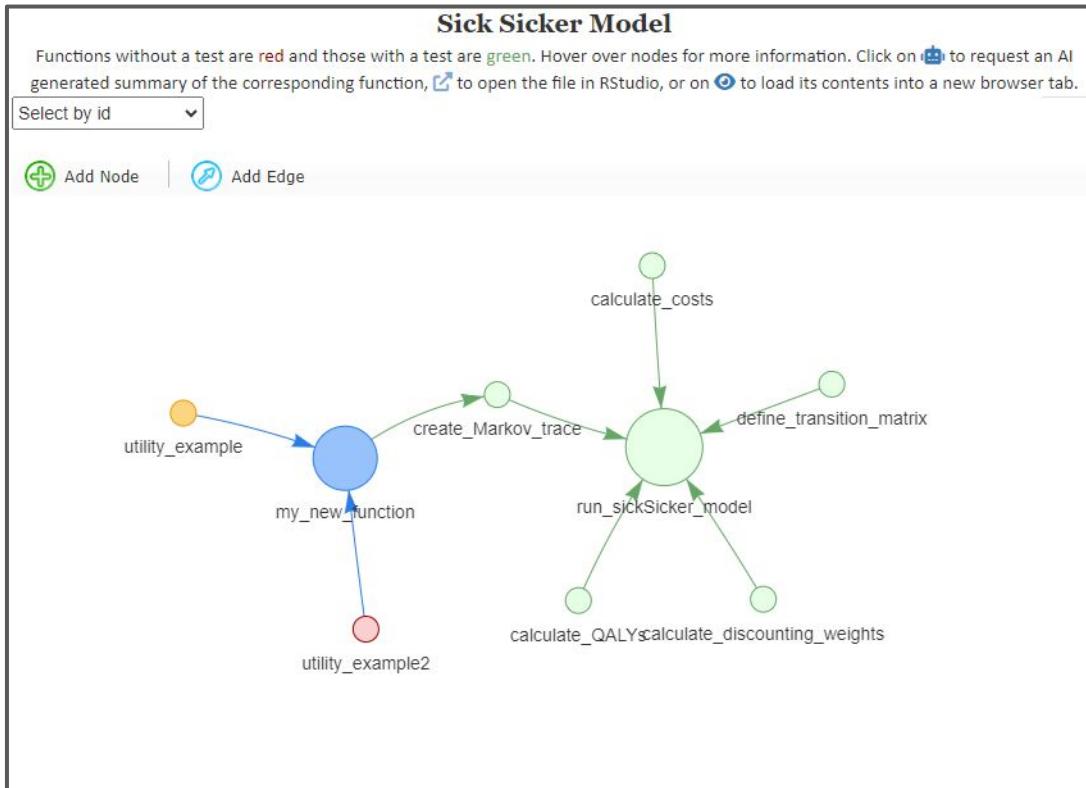
`create_Markov_trace` 

`create_Markov_trace`    

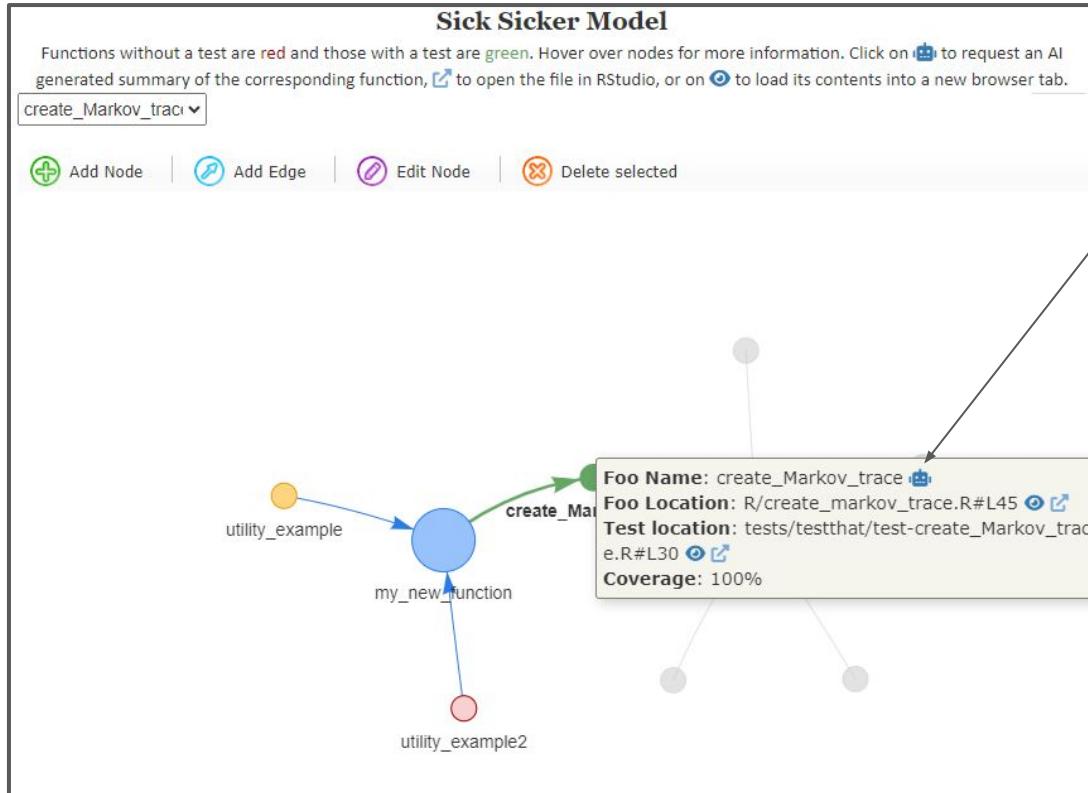
Foo Name: `create_Markov_trace` 
Foo Location: `R/create_markov_trace.R#L45`  
Test location: `tests/testthat/test-create_Markov_trace.R#L30`  
Coverage: 100%



assertHE model reviewer

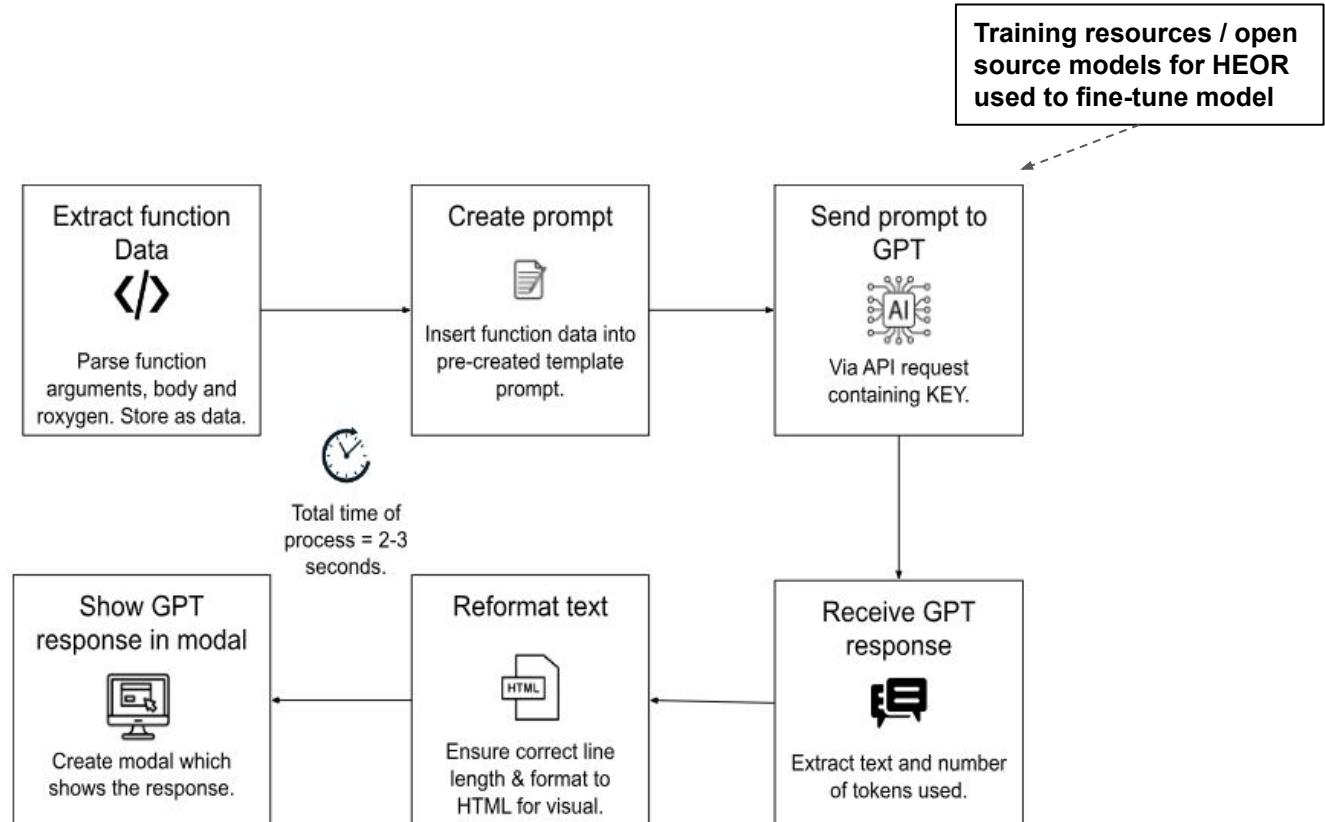


assertHE model reviewer

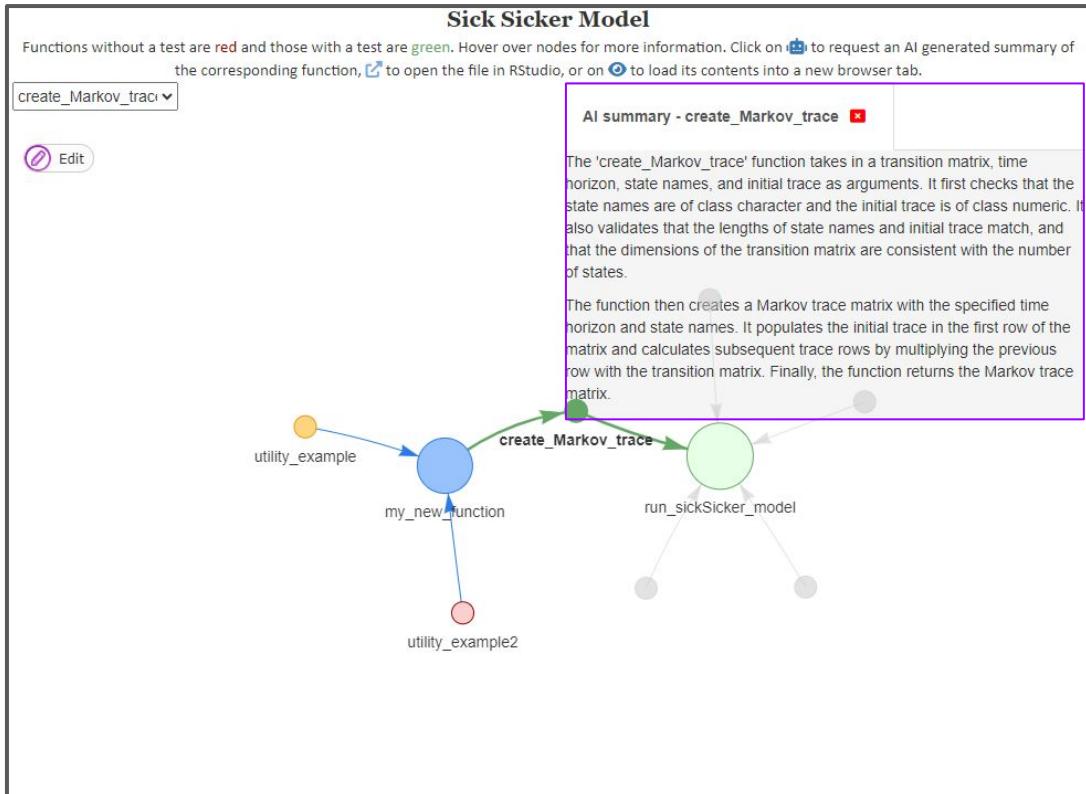


Generate LLM summary of function.

assertHE model reviewer



assertHE model reviewer





Case Studies



Case Studies

We have used the assertHE package on several models as test cases:

- [NICE RCC Model](#)
- [sicksickerPack](#) teaching model contained in a package.
- [cdx2cea](#) as described in [Alarid-Escudero et al. 2022](#)
- [DOACs-AF-Economic-model](#) developed by Bristol University
- The CGD AMR Cost model - in press.
- [Embedding Economics Analysis](#) Diabetes Microsimulation model described in (in press).

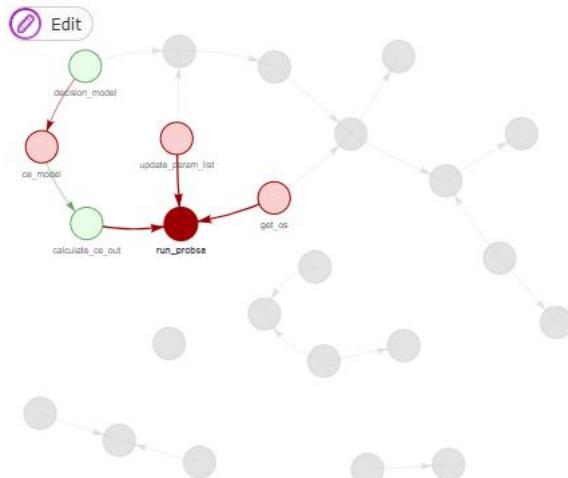
Others have used assertHE on their own models that are not in the public domain. We welcome this. Please get in contact if you have any issues or suggestions for improvements.

Case Study

Function Network

Functions without a test are **red** and those with a test are **green**. Hover over nodes for more information. Click on  to request an AI generated summary of the corresponding function,  to open the file in RStudio, or on  to load its contents into a new browser tab.

run_probsa



AI summary - run_probsa

```html

The '`run_probsa`' function runs probabilistic sensitivity analysis (PSA) on a given input dataset. If the '`parallel`' argument is set to `TRUE`, the function parallelizes the PSA process using multiple cores based on the operating system. It then calculates costs and effects for each simulation, aggregates the results, and returns them in separate data frames.

If the '`parallel`' argument is set to `FALSE`, the function runs the PSA simulations in series. It iterates through each simulation, updates parameters, calculates costs and effects, and prints the progress. Finally, it aggregates the results and returns them in separate data frames. The function returns a list containing the costs and effects data frames.

...



**ScienceDirect**

Contents lists available at sciencedirect.com  
journal homepage: www.elsevier.com/locate/jeud

Economic Evaluation

**CDX2 Biomarker Testing and Adjuvant Therapy for Stage II Colon Cancer: An Exploratory Cost-Effectiveness Analysis**

Fernando Alvaro Escudero, PhD, Deborah Schrag, MD, MPH, Karen M. Kornblith, ScD

#### ABSTRACT

**Objective:** Adjuvant chemotherapy is not recommended for patients with average-risk stage II (T1N0) colon cancer. Nevertheless, a subgroup of these patients who are CDX2-negative might benefit from adjuvant chemotherapy. We evaluated the cost-effectiveness of testing for the absence of CDX2 expression followed by adjuvant chemotherapy versus no testing for patients with average-risk stage II colon cancer.

**Methods:** We developed a decision model to evaluate the hypothesis that 65-year-old patients with average-risk stage II colon cancer with 72% of these patients being CDX2-negative under 2 different interventions: [1] test for the absence of CDX2 expression followed by adjuvant chemotherapy, or [2] no testing followed by adjuvant chemotherapy. We used a Markov model to determine the incremental cost-effectiveness ratio (ICER) of testing for CDX2 followed by adjuvant chemotherapy for any patient. We derived disease progression parameters, adjuvant chemotherapy effectiveness and utilities from published analyses, and cancer cost rates from the Surveillance, Epidemiology, and End Results (SEER) Medicare database.

**Results:** Testing for CDX2 followed by FOLOX for CDX2-negative patients had an incremental cost-effectiveness ratio of \$10,000/QALY vs \$10,991/QALY for no testing followed by FOLOX. Testing for CDX2 followed by FOLOX for CDX2-negative patients remains cost-effective for patients with an ICER of \$10,000/QALY. Testing for CDX2 followed by FOLOX for CDX2-negative patients remains cost-effective for patients with an ICER of \$25,000/QALY.

**Conclusion:** Testing of patients with stage II colon cancer for CDX2 and administration of adjuvant treatment to the subgroup of CDX2-negative is a cost-effective and high-value management strategy across a broad range of plausible assumptions.

**Keywords:** CDX2, cost-effectiveness analysis, decision-analytic model, immunohistochemical testing, stage II colon cancer.

*JOURNAL OF CLINICAL ONCOLOGY*, Vol. 20, No. 18, October 1, 2002, pp. 409–416

#### Introduction

Adjuvant chemotherapy is not recommended for patients with average-risk stage II (T1N0) colon cancer,<sup>1–3</sup> and thus, these patients are usually treated with surgery alone.<sup>4</sup> Nevertheless, a recent study by Dalakas et al<sup>5</sup> described a small subgroup of patients with stage II colon cancer who were positive for the CDX2 transcription factor that associated with clinical benefit from adjuvant chemotherapy. CDX2 is a master transcription factor that is expressed in the majority of cells of a candidate biomarker of mature colon epithelial tissues. In this study, the authors found that CDX2 expression was able to conduct a systematic search for a biomarker to identify undifferentiated tumors in a collection of human colon gene expression array experiments. The authors found that CDX2 expression was associated with tumor aggressiveness and was able to predict benefit from adjuvant chemotherapy to a subgroup most likely to benefit. This study aims to quantify the long-term benefits, costs, and cost-effectiveness of testing average-risk patients with stage II colon cancer for the absence of CDX2 biomarker expression followed

National Cancer Institute, which were analyzed for CDX2 expression by immunohistochemical (IHC) analysis. Among all tumors analyzed in the validation data set, 48 of 669 (7.2%) were CDX2-negative, defining the CDX2-negative population. The validation data set showed a median survival of 10 years for patients with CDX2 expression in a minority of malignant epithelial cells,<sup>5,6</sup> The study also showed that CDX2-negative patients had poorer overall survival than CDX2-positive patients, and the 5-year DFS was greater for the CDX2-negative patients who received adjuvant chemotherapy compared to those who did not receive adjuvant chemotherapy. The ability to test average-risk patients with stage II colon cancer for the absence of CDX2 expression to target adjuvant chemotherapy to a subgroup most likely to benefit could reduce colon cancer mortality and minimize adjuvant chemotherapy.

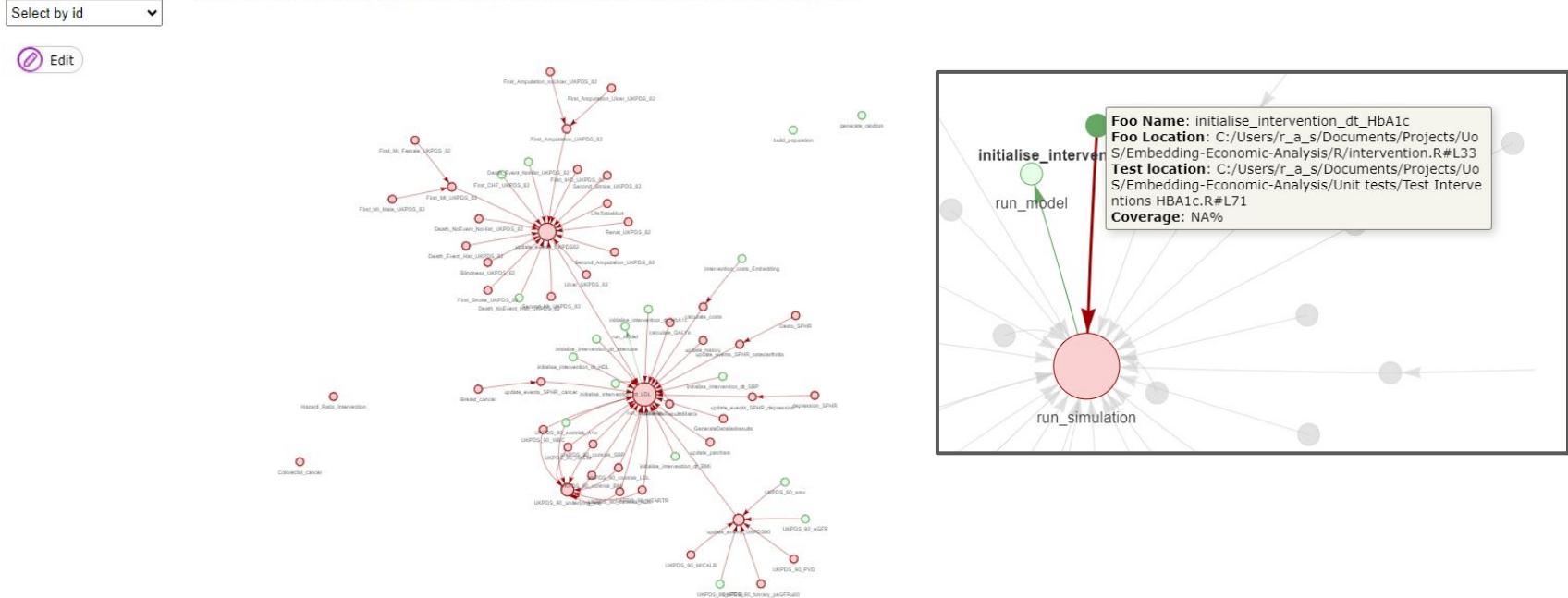
This study aims to quantify the long-term benefits, costs, and cost-effectiveness of testing average-risk patients with stage II colon cancer for the absence of CDX2 biomarker expression followed

© 2002 American Society of Clinical Oncology. Published by Lippincott Williams & Wilkins.

# Case Study: Embedding Economic Analysis

## Embedding-Economic-Analysis Repository

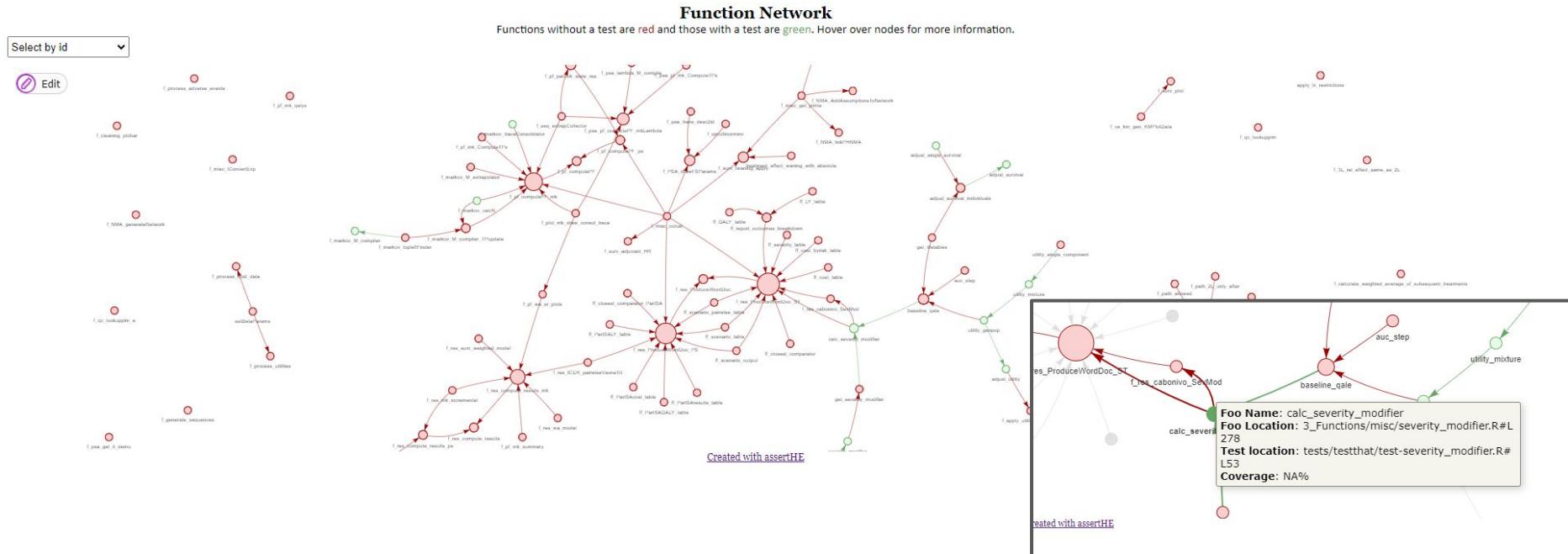
Functions without a test are red and those with a test are green. Hover over nodes for more information.



Created with assertHE

<https://github.com/DanPollardSheff/Embedding-Economic-Analysis>

# Case Study: NICE RCC Pilot





# Next steps

- Welcome contributions
  - Testing the software on your models
  - Suggesting improvements (see contribution page on GitHub)
  - Expansion of testing functionality
- Future development:
  - LLM Chatbot integration (using DPA teaching material to fine-tune).
  - Language selection (in progress)
- Open access publication imminent
- CRAN submission (Autumn '24)
- **Continued open-access development to maintain a collaborative tool**

Building Health Economic Models in R

Search

Table of contents

1 Using this book

2 About the authors

3 Background

4 Introduction to R

5 Version Control

6 Intermediate R

7 Wrangling with data in R

8 Partitioned Survival Models

9 Cohort State Transition Models

10 Automated Reporting

11 Shiny for Health Economics

12 Advanced Data Visualisation

13 R Packages

14 Further Resources

References

## 1 Using this book

**Note from the authors:** This work is a living document and is being adapted all the time based upon comments. New sections are being included prior to teaching. If you have suggestions for improving this book, please contact Robert Smith by email: [rsmith@darkpeakanalytics.com](mailto:rsmith@darkpeakanalytics.com).

— Dark Peak Analytics Teaching Team

This book was created to provide additional support for the taught course. It includes all of the code chunks, exercises and solutions which we cover in the taught sessions. The book serves as a first point of reference, and directs the reader to additional resources. The book is written in R using the `bookdown` package, which converts each chapter from an Rmarkdown file into a PDF or HTML book. Later in the course, we will cover how this process works.

There are currently thirteen chapters in the book, as shown in the table below. The content that is visible to you will depend on the scope of the course that you are enrolled on. For example those taking a course on the use of shiny for Cohort State Transition Models may have access to Chapter 7 and 10, whereas those doing an introductory R course may have access to chapter 1 only.

<https://github.com/dark-peak-analytics/assertHE>

# Open Access Publication

1  
2  
3 **assertHE**: an R package to improve quality assurance of health  
4 economic models  
5  
6 Wellcome Open Research  
7

## 8 Authors

9 Robert A Smith (Dark Peak Analytics, University of Sheffield)  
10 Yevgeniy Samyshkin (GSK),  
11 Wael Mohammed (Dark Peak Analytics, University of Sheffield)  
12 Tom Ward (GSK)  
13 Alan Martin (GSK)  
14 Sarah-Jane Anderson (GSK)  
15 Paul Schneider (Dark Peak Analytics, University of Sheffield)  
16 Baris Deniz (GSK)  
17 Dawn Lee (University of Exeter)  
18 Prof. Gianluca Baio (University College London)  
19 Howard Thom (University of Bristol)  
20 Nathan Green (University College London)  
21 Felicity Lamrock ([Queens University Belfast](#))  
22 Brett McQueen (University of Colorado at Denver)  
23 Marina Richardson (Institute for Clinical and Economic Review)  
24 Mohamed El Alili (Zorginstituut Nederland)  
25 Xavier Pouwels (University of Twente).  
26 March 2023  
27

## 28 Abstract

29

30 **Background:** Health economic evaluation models are increasingly used to inform decisions about the  
31 allocation of health care resources. Ensuring the robustness, reliability, and reproducibility of these  
32 models is critical. Currently, quality assurance is conducted by experts assessing the different  
33 components of the model manually in isolation and in combination. However, this is resource  
34 intensive. Understanding how the different components of the model fit together is time consuming,  
35 and testing each part of the model is sometimes not feasible under the timescales provided to  
36 reviewers. To aid in this, we propose the *assertHE* R package.

## 37 Methods:

38 The open source *assertHE* package provides testing functionality for modellers and reviewers of  
39 health economic models. It provides a series of common checks, which can be integrated into the  
40 model development workflow to reduce the probability of common errors. It also provides a suite of  
41 functions which allow users to better understand the network of algorithms (functions) contained in the  
42 model, where they are defined, if (and where) they are tested, and the test coverage of those that  
43 have.

44 **Results:** We applied the *assertHE* package to two health economic models, showing how to include  
45 the check functions within the model code and showing how to visualise a network of functions, see  
46 the test coverage, and obtain a Generative Pretrained Transformer (GPT) Large Language Model  
47 (LLM) generated summary of any function in the codebase. We have worked with collaborators from  
48 industry, regulators and academia to develop the package to be applicable to the widest possible  
49 range of models, making adaptations to the source code based upon feedback.

50 **Conclusions:** The *assertHE* R software package offers a toolkit for health economists building and  
51 reviewing models, facilitating a more robust and efficient quality assurance process. We hope this will  
52 ultimately improve the quality, transparency and efficiency of the health economic evaluation process  
53 for models built in R.

54

## 55 Key Words:

56 *R, Health Economics, Unit Testing, Model Validation*

57

58

<https://drive.google.com/file/d/1ZR0zMZjiEERdzoQM49Pm2LXtOSe4agMi/view?usp=sharing>

# Q & A





# Book topics

We run courses on several topics relating to building health economic models in R.

- Introduction to R
- Version Control
- Intermediate R
- Wrangling with data in R
- Partitioned Survival Models
- State Transition Models
- Efficient Microsimulation in R
- Automated Reporting
- RShiny for Health Economics
- Advanced Data visualisation
- R packages
- Reviewing Health Economic Models in R

[Building Health Economic Models in R](#)

Search

Table of contents

[1 Using this book](#)

[2 About the authors](#)

[3 Background](#)

[4 Introduction to R](#)

[5 Version Control](#)

[6 Intermediate R](#)

[7 Wrangling with data in R](#)

[8 Partitioned Survival Models](#)

[9 Cohort State Transition Models](#)

[10 Automated Reporting](#)

[11 R shiny for Health Economics](#)

[12 Advanced Data Visualisation](#)

[13 R Packages](#)

[14 Further Resources](#)

References

## 1 Using this book

**Note from the authors:** This work is a living document and is being adapted all the time based upon comments. New sections are being included prior to teaching. If you have suggestions for improving this book, please contact Robert Smith by email: [rsmith@darkpeakanalytics.com](mailto:rsmith@darkpeakanalytics.com).

— Dark Peak Analytics Teaching Team

This book was created to provide additional support for the taught course. It includes all of the code chunks, exercises and solutions which we cover in the taught sessions. It serves as a first point of reference, and directs the reader to additional resources. The book is written in R using the `bookdown` package, which converts each chapter from an Rmarkdown file into a PDF or HTML book. Later in the course, we will cover how this process works.

There are currently thirteen chapters in the book, as shown in the table below. The content that is visible to you will depend on the scope of the course that you are enrolled on. For example those taking a course on the use of shiny for Cohort State Transition Models may have access to Chapter 7 and 10, whereas those doing an introductory R course may have access to chapter 1 only.

[Bespoke training courses](#)



# Making Health Economic Models Shiny: Sept 24



## Dates

The online course sessions are held on four consecutive Thursdays in September and October 2024:

1. Thursday, 12 September 2024
2. Thursday, 19 September 2024
3. Thursday, 26 September 2024
4. Thursday, 03 October 2024

Each session runs from:

13:00 - 16:00 GMT (London time)  
08:00 - 11:00 EST (New York time)  
17:00 - 20:00 GST (Dubai time)

**PLUS:** optional drop-in code clinics are held on Tuesdays:

1. Tuesday, 17 September 2024
2. Tuesday, 24 September 2024
3. Tuesday, 01 October 2024
4. Tuesday, 08 October 2024

Each code clinic runs from:

13:00 - 14:30 GMT (London time)  
08:00 - 09:30 EST (New York time)  
17:00 - 18:30 GST (Dubai time)

<https://www.courses.darkpeakanalytics.com/>



# Further resources

**Smith, R.A.**, and Schneider, P.P. (2020). Making health economic models Shiny: A tutorial. *Wellcome Open Res*, 5, 69. <https://doi.org/10.12688/wellcomeopenres.15807.2>

**Smith, R.A.**, Schneider, P.P., and Mohammed, W. (2022). Living HTA: Automating Health Economic Evaluation with R. *Wellcome Open Res*, 7, 194. <https://doi.org/10.12688/wellcomeopenres.17933.2>

**Smith, R.A.**, Mohammed, W. and Schneider, P.P. (2023). R Packages for health economic evaluation: A tutorial. *Wellcome Open Res* (In Press).

Thokala, P., Srivastava, T., **Smith, R.**, Ren, S., Whittington, M.D., Elvidge, J., Wong, R., Uttley, L. (2023). Living Health Technology Assessment – Issues, Challenges and Opportunities. *Pharmacoeconomics*.

Other Publications: [Google Scholar](#)

Open Source code for Shiny: [https://github.com/RobertASmith/healthecon\\_shiny](https://github.com/RobertASmith/healthecon_shiny)

Open Source code for Living HTA: <https://github.com/RobertASmith/plumberHE>

Open Source code for Packaging: <https://github.com/dark-peak-analytics/sicksickerPack>

Other open source code: <https://github.com/dark-peak-analytics>

Rob's talks: <https://github.com/RobertASmith/talks>

Rob's CV: [CV](#)